



**ESCUELA SUPERIOR DE INGENIERÍA**

**INGENIERO TÉCNICO EN INFORMÁTICA DE**  
**GESTIÓN**

**SERVICIO WEB PARA LA VISUALIZACIÓN GRÁFICA**  
**DE LAS CONTRIBUCIONES EN WEBS**  
**COLABORATIVAS**

Francisco José Benítez Pérez

23 de julio de 2011





## ESCUELA SUPERIOR DE INGENIERÍA

### INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

#### SERVICIO WEB PARA LA VISUALIZACIÓN GRÁFICA DE LAS CONTRIBUCIONES EN WEBS COLABORATIVAS

- Departamento: Lenguajes y Sistemas Informáticos
- Director del proyecto: Juan Manuel Dodero Beardo
- Autor del proyecto: Francisco José Benítez Pérez

Puerto Real, 23 de julio de 2011

Fdo: Francisco José Benítez Pérez





# Índice general

Índice general	v
Índice de figuras	IX
Índice de tablas	XIII
<b>1 Introducción</b>	<b>1</b>
1.1. Objetivos . . . . .	1
1.2. Motivación . . . . .	3
1.3. Alcance . . . . .	3
1.3.1. Orígenes de los datos . . . . .	4
1.3.2. Motores de visualización gráfica . . . . .	4
1.4. Metodología de desarrollo . . . . .	5
1.5. Planificación . . . . .	6
1.5.1. Objetivo inicial . . . . .	6
1.5.2. División del trabajo . . . . .	6
1.5.3. Estimación de tiempos . . . . .	7
1.5.4. Identificación de hitos . . . . .	8
1.5.5. Encadenamiento de actividades . . . . .	9
1.5.6. Planificación temporal . . . . .	10
1.6. Coste de desarrollo . . . . .	11
<b>2 Tecnologías usadas</b>	<b>13</b>
2.1. Entornos y herramientas de desarrollo . . . . .	13
2.2. Modelado orientado a objetos . . . . .	15
2.3. Lenguajes y frameworks de desarrollo software . . . . .	16
2.3.1. Prefuse . . . . .	16
2.3.2. Gephi . . . . .	17
2.3.3. History Flow . . . . .	19
2.3.4. Modos de visualización . . . . .	21
2.3.5. Restlet . . . . .	23
2.3.5.1. Características de Restlet . . . . .	24
2.3.5.2. Arquitectura del Framework . . . . .	25
2.3.6. Java . . . . .	26

2.3.7.	Lenguaje UML . . . . .	27
2.3.7.1.	Antecedentes UML . . . . .	27
2.3.7.2.	Características UML . . . . .	27
<b>3</b>	<b>Especificaciones generales del proyecto</b>	<b>29</b>
3.1.	Parámetros y funciones incluidas en WebHistoryFlow . . . . .	29
3.2.	Especificaciones preliminares . . . . .	30
3.3.	Restricciones generales . . . . .	30
3.4.	Suposiciones y dependencias . . . . .	31
3.5.	Futuras versiones . . . . .	32
<b>4</b>	<b>Desarrollo general del proyecto</b>	<b>33</b>
4.1.	Análisis . . . . .	33
4.1.1.	Identificación actores y casos de uso . . . . .	33
4.1.2.	Modelado de casos de uso . . . . .	35
4.1.2.1.	Diagrama de casos de uso . . . . .	35
4.1.2.2.	Descripción casos de uso . . . . .	36
4.1.3.	Modelado conceptual de datos . . . . .	49
4.1.3.1.	Diagrama de clases conceptuales . . . . .	49
4.1.4.	Modelo de comportamiento del sistema . . . . .	53
4.1.4.1.	Diagramas de secuencia del sistema (DSS) y contrato de las operaciones . . . . .	53
4.2.	Diseño . . . . .	66
4.2.1.	Diagramas de interacción o colaboración . . . . .	66
4.2.1.1.	Diagrama de colaboración CU Visualización de Instrucciones de Uso . . . . .	66
4.2.1.2.	Diagrama de colaboración de CU con formato de salida History Flow artículo wiki . . . . .	67
4.2.1.3.	Diagrama de colaboración de CU con formato de salida History Flow fichero en svn . . . . .	71
4.2.1.4.	Diagrama de colaboración de CU con formato de salida Prefuse artículo wiki . . . . .	74
4.2.1.5.	Diagrama de colaboración de CU con formato de salida Prefuse fichero en svn . . . . .	77
4.2.1.6.	Diagrama de colaboración de CU con formato de salida Gephi artículo wiki . . . . .	79
4.2.1.7.	Diagrama de colaboración de CU con formato de salida Gephi fichero en svn . . . . .	82
4.2.2.	Diseño de clases y asociaciones . . . . .	84
4.3.	Implementación . . . . .	89
4.3.1.	Características de la implementación . . . . .	89
4.3.2.	Trabajar con el framework Restlet . . . . .	91
4.3.2.1.	Introducción a Restlet . . . . .	91
4.3.2.2.	Gestión de Recursos . . . . .	92

4.3.2.3.	Componentes y conectores . . . . .	93
4.3.2.4.	La clase raíz Restlet . . . . .	93
4.3.2.5.	Fichero web.xml para Restlet . . . . .	94
4.3.2.6.	Implementar las rutas en Restlet . . . . .	95
4.3.3.	Características comunes para recursos History Flow, Prefuse y gephi . . . . .	98
4.3.4.	Adaptación, implementación History Flow a WebHistoryFlow . . . . .	99
4.3.5.	Implementar Prefuse y Gephi en WebHistoryflow . . . . .	101
4.3.6.	Interpretación de la representación gráfica de Prefuse y Gephi . . . . .	103
4.3.6.1.	Interpretación gráfico Prefuse . . . . .	103
4.3.6.2.	Interpretación gráfico Gephi . . . . .	106
<b>5</b>	<b>Evaluación y pruebas</b>	<b>109</b>
5.1.	Definiciones para los casos de prueba . . . . .	109
5.2.	Diseños para los casos de prueba . . . . .	110
5.3.	Casos de prueba . . . . .	111
5.3.1.	Diseño de los casos de prueba según el estándar IEEE Std. 829 . . . . .	112
5.3.1.1.	Plan de pruebas . . . . .	112
5.3.1.2.	Especificación de diseño de pruebas . . . . .	116
5.3.1.3.	Especificación de casos de prueba . . . . .	119
5.3.1.4.	Especificación de procedimientos de prueba . . . . .	149
5.4.	Utilización de recursos WebHistoryFlow en implementaciones cliente . . . . .	150
<b>6</b>	<b>Conclusiones</b>	<b>155</b>
<b>7</b>	<b>Apéndice A</b>	<b>159</b>
7.1.	Definiciones, acrónimos y abreviaturas . . . . .	159
<b>8</b>	<b>Apéndice B</b>	<b>163</b>
8.1.	Guía de usuario . . . . .	163
	<b>Bibliografía</b>	<b>169</b>



# Índice de figuras

1.1. Metodología de desarrollo de un Modelo Lineal Secuencial. . . . .	5
1.2. Estructura desglosada descomposición trabajo. . . . .	7
1.3. Módulo Bibliografía. . . . .	7
1.4. Módulo Recursos. . . . .	7
1.5. Módulo APIs externas. . . . .	8
1.6. Estimación de tiempos del proyecto. . . . .	8
1.7. Hitos para la consecución del objetivo final. . . . .	9
1.8. Red de actividades previstas. . . . .	10
1.9. Diagrama de Gantt PFC - relación temporal actividades WebHistoryFlow. . . . .	11
2.1. Estructura de visualización del framework de Prefuse. . . . .	17
2.2. Aplicación de escritorio de Gephi. . . . .	18
2.3. Creación primera versión en una wiki, <i>ver.1</i> . . . . .	19
2.4. Creación y modificación de la ultima versión, <i>ver.2</i> . . . . .	20
2.5. Eliminación y modificación de la última versión, <i>ver.3</i> . . . . .	20
2.6. Inserción en la última versión, <i>ver.4</i> . . . . .	20
2.7. Conexión de versiones a través de paralelogramos. . . . .	21
2.8. Representación de versiones en modo grupo. . . . .	21
2.9. Representación de versiones en modo autor individual. . . . .	22
2.10. Representación de versiones en modo antigüedad del flujo de datos. . . . .	22
2.11. Representación de versiones en modo cambios recientes en cada versión. . . . .	23
2.12. Representación de versiones en modo links de comienzo en cada versión. . . . .	23
2.13. Contenido Paquete API Restlet. . . . .	24
2.14. Plataformas soportadas por el framework de Restlet. . . . .	25
2.15. Arquitectura framework Restlet. . . . .	26
2.16. Plataforma Java 2. . . . .	26
2.17. Vista general de elementos UML. . . . .	28
4.1. Casos de Uso pertenecientes al Sistema History Flow. . . . .	35
4.2. Diagrama de clases. . . . .	49

4.3. Diagrama de clases, bloque superior. . . . .	50
4.4. Diagrama de clases, bloque inferior. . . . .	52
4.5. DSS, Caso uso: Visualización de instrucciones. . . . .	53
4.6. DSS, Caso uso: Solicitar Gráfico History Flow Artículo Wiki. . . .	54
4.7. DSS, Caso uso: Solicitar Gráfico History Flow Artículo Wiki de un Autor. . . . .	55
4.8. DSS, Caso uso: Solicitar Gráfico Gephi Artículo Wiki. . . . .	56
4.9. DSS, Caso uso: Solicitar Datos Formato Gephi Artículo Wiki. . . .	57
4.10. DSS, Caso uso: Solicitar Gráfico Prefuse Artículo Wiki. . . . .	58
4.11. DSS, Caso uso: Solicitar Datos Formato Prefuse Artículo Wiki. . .	59
4.12. DSS, Caso uso: Solicitar Gráfico History Flow Fichero Subversión.	60
4.13. DSS, Caso uso: Solicitar Gráfico History Flow Fichero Subversión de un Autor. . . . .	61
4.14. DSS, Caso uso: Solicitar Gráfico Gephi Fichero Subversión. . . . .	62
4.15. DSS, Caso uso: Solicitar Datos Formato Gephi Fichero Subversión.	63
4.16. DSS, Caso uso: Solicitar Gráfico Prefuse Fichero Subversión. . . .	64
4.17. DSS, Caso uso: Solicitar Datos Formato Prefuse Fichero Subversión.	65
4.18. Diagrama de interacción, C.U.: Visualización Instrucciones de Uso	66
4.19. Diagrama de interacción, C.U.: Solicitar Gráfico History Flow Ar- tículo Wiki, bloque superior figura 4.22 . . . . .	67
4.20. Diagrama de interacción, C.U.: Solicitar Gráfico History Flow Ar- tículo Wiki, bloque medio figura 4.22 . . . . .	68
4.21. Diagrama de interacción, C.U.: Solicitar Gráfico History Flow Ar- tículo Wiki, bloque inferior figura 4.22 . . . . .	69
4.22. Diagrama de interacción, C.U.: Solicitar Gráfico History Flow Ar- tículo Wiki . . . . .	70
4.23. Diagrama de interacción, C.U.: Solicitar Gráfico History Flow de un fichero de una svn, bloque inferior. . . . .	72
4.24. Diagrama de interacción, C.U.: Solicitar Gráfico History Flow de un fichero de una svn. . . . .	73
4.25. Diagrama de interacción, C.U.: Solicitar Gráfico Prefuse de un ar- tículo wiki, bloque inferior. . . . .	74
4.26. Diagrama de interacción, C.U.: Solicitar Gráfico Prefuse de un ar- tículo wiki. . . . .	75
4.27. Diagrama de interacción, C.U.: Solicitar Gráfico Prefuse de un fi- chero de una svn, bloque inferior. . . . .	77
4.28. Diagrama de interacción, C.U.: Solicitar Gráfico Prefuse de un fi- chero de una svn. . . . .	78
4.29. Diagrama de interacción, C.U.: Solicitar Gráfico Gephi de un ar- tículo wiki, bloque inferior. . . . .	79
4.30. Diagrama de interacción, C.U.: Solicitar Gráfico Gephi de un ar- tículo wiki. . . . .	80
4.31. Diagrama de interacción, C.U.: Solicitar Gráfico Gephi de un fichero de una svn, bloque inferior. . . . .	82

4.32. Diagrama de interacción, C.U.: Solicitar Gráfico Gephi de un fichero de una svn. . . . .	83
4.33. Diseño de clases y asociaciones. . . . .	84
4.34. Diseño de clases y asociaciones, bloque superior. . . . .	85
4.35. Diseño de clases y asociaciones, bloque inferior. . . . .	87
4.36. IDE, Interfaz de Desarrollo Integrado, Netbeans 7.0 . . . . .	90
4.37. Ejecución del Servidor Virtual de Restlet. . . . .	91
4.38. Unión entre un Identificador, un Recurso y una Representación. . .	92
4.39. Comunicación entre componentes. . . . .	94
4.40. Herencia de clases del paquete raíz. . . . .	94
4.41. Fichero web.xml, usando clase Component. . . . .	95
4.42. Fichero web.xml, usando clase Application. . . . .	95
4.43. Clase Application para WebHistoryFlow, agrupa recursos. . . . .	97
4.44. Recurso que devuelve un representación para un tipo de medio HTML. . . . .	98
4.45. Captura de pantalla del servidor, creando y borrando ficheros temporales. . . . .	98
4.46. Representación gráfico applet demo Prefuse. . . . .	104
4.47. Representación gráfico pdf demo Gephi. . . . .	107
5.1. Ciclo completo de las pruebas [13]. . . . .	110
5.2. Ciclo pruebas estándar IEEE 829. . . . .	111
5.3. Recurso : Página Web información uso aplicación WebHstoryFlow. .	119
5.4. Recurso : Gráfico artículo wiki con formato History Flow, modo: <i>grupo</i> . . . . .	121
5.5. Recurso : Gráfico artículo wiki con formato History Flow, modo: <i>link</i> . .	121
5.6. Recurso : Gráfico artículo wiki con formato History Flow, modo: <i>tiempo</i> . . . . .	122
5.7. Recurso : Gráfico artículo wiki con formato History Flow, modo: <i>reciente</i> . . . . .	122
5.8. Recurso : Gráfico artículo wiki con formato History Flow de un autor, modo: <i>individual</i> . . . . .	123
5.9. Recurso : Gráfico fichero svn con formato History Flow, modo: <i>grupo</i> . .	126
5.10. Recurso : Gráfico fichero svn con formato History Flow, modo: <i>link</i> . .	126
5.11. Recurso : Gráfico fichero svn con formato History Flow, modo: <i>tiempo</i> . .	127
5.12. Recurso : Gráfico fichero svn con formato History Flow, modo: <i>reciente</i> . . . . .	127
5.13. Recurso : Gráfico formato History Flow de un de los autores del fichero perteneciente a una svn, modo: <i>individual</i> . . . . .	128
5.14. Muestra de directorio de un fichero en <i>www.assembla.com</i> . . . . .	129
5.15. Recurso : Gráfico artículo wiki con formato Gephi. . . . .	131
5.16. Recurso : Gráfico formato Gephi de un fichero perteneciente a una svn. . . . .	133

5.17. Recurso : Fichero de datos de un artículo wiki con salida en formato Gephi, vista nodos. . . . .	135
5.18. Recurso : Fichero de datos de un artículo wiki con salida en formato Gephi, vista aristas. . . . .	135
5.19. Recurso : Fichero de datos de un fichero de una svn con salida en formato Gephi, vista nodos. . . . .	137
5.20. Recurso : Fichero de datos de un fichero de una svn con salida en formato Gephi, vista aristas. . . . .	137
5.21. Recurso : Gráfico applet de artículo wiki con formato Prefuse. . . .	139
5.22. Recurso : Panel modificación vista del applet con formato Prefuse. .	139
5.23. Recurso : Gráfico applet de fichero de una svn con formato Prefuse. .	141
5.24. Recurso : Gráfico fichero datos XML de un artículo wiki con formato Prefuse, vista nodos. . . . .	142
5.25. Recurso : Gráfico fichero datos XML de un artículo wiki con formato Prefuse, vista aristas. . . . .	143
5.26. Recurso : Gráfico fichero datos XML formato Prefuse de fichero de una svn, vista nodos. . . . .	144
5.27. Recurso : Gráfico fichero datos XML formato Prefuse de fichero de una svn, vista aristas. . . . .	145
5.28. Recurso : Mensaje error en modo: <i>individual</i> , sin indicar autor. . .	146
5.29. Recurso : Mensaje error Restlet cuando no encuentra coincidencias en rutas. . . . .	147
5.30. Recurso : Mensaje error cuando no existe la opción en las rutas definidas. . . . .	148
5.31. Gestor de Contenidos Joomla solicitando recurso gráfico a WebHistoryFlow. . . . .	150
5.32. Uso de WebHistoryFlow en una página web de Joomla. . . . .	151
5.33. Gestor de Contenidos Joomla, edición de código HTML del artículo. .	151
5.34. Ventana JFrame de Java con recurso WebHistoryFlow. . . . .	152
5.35. Ejemplo implementación Java para uso de WebHistoryFlow. . . . .	152
5.36. Uso de fichero <i>.gml</i> creado con WebHistoryFlow en Aplicación Escritorio Gephi 7.0 . . . . .	153
5.37. Datos <i>.gml</i> creado con WebHistoryFlow en Aplicación Escritorio Gephi 7.0 . . . . .	153



# Índice de tablas

1.1.	Algunos Tipos de Medios estándar soportados por Restlet . . . . .	3
1.2.	Relación actividades Diagrama de Gantt para la <i>figura</i> 1.9. . . . .	11
1.3.	Relación de costes para el desarrollo del proyecto. . . . .	12
8.1.	Formatos predefinidos en artículos mediawiki para WebHistoryFlow.	164
8.2.	Ejemplos solicitud recursos WebHistoryFlow para artículos media-wiki. . . . .	164
8.3.	Formatos predefinidos para ficheros subversiones en WebHistoryFlow.	165
8.4.	Ejemplos solicitud recursos WebHistoryFlow para ficheros de sub-versiones. . . . .	166



# Capítulo 1

## Introducción

### 1.1. Objetivos

El objetivo principal de este proyecto consiste en la implementación de un *Servicio Web* a través de la parametrización de URI's, a partir de la cual, se obtendrán representaciones gráficas partiendo de contribuciones realizadas en webs colaborativas.

Se enfocará el *Servicio Web* utilizando un conjunto de protocolos y estándares para intercambio de datos cliente-servidor, buscando la interoperabilidad entre distintas aplicaciones software, independientemente de la plataforma que utilicen los clientes. Este nuevo *Servicio Web* a partir de ahora, será referenciado como, *WebHistoryFlow*.

Teniendo en cuenta este objetivo general, y con la necesidad de concretar objetivos, será necesario desglosarlo y tratarlo a través de unos objetivos mas específicos como los relacionados a continuación.

- Proporcionar un *Servicio Web* de la aplicación History Flow a través del servicio online WebHistoryFlow.
- Aplicar una metodología de trabajo basada en una arquitectura REST [4] para la realización del proyecto.
- Definir un esquema de peticiones URI <sup>1</sup>. Este esquema estaría construido a partir de un framework llamado Restlet [5] para conseguir mejorar la interoperabilidad del servicio de visualización gráfica.

Estas peticiones se desglosarían en función de unos determinados tipos de ficheros y de sus correspondientes posibles salidas para dar servicio a:

---

<sup>1</sup>Identificador uniforme de recurso. Identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, etc...). Formado por: Esquema,Autoridad, Ruta, Consulta y Fragmento.

- Solicitudes de recursos para obtener archivos con formato Media-Wiki, *wikis*.
- Solicitudes de recursos para procesar un fichero perteneciente a subversiones (svn).
- Recuperar datos a partir de fuentes de webs colaborativas, importando sus datos de unos plugins específicos para ello. Plugins como el suministrado por History Flow para importar datos de una wiki o como el suministrado por Richard Fearn [25], para importar las distintas versiones de un fichero de una subversión (svn) <sup>2</sup>.
- Establecer unos recursos con salidas distintas a la tradicional en formato HTML, a través de solicitudes al protocolo HTTP con Distintos tipos de contenidos, definidos en función de un nombre el cual definirá su forma de interpretación conforme a representaciones de tipos de medios estándar.

Tanto para wikis como para subversiones suministraría salidas para las siguientes API's que proporcionan representaciones gráficas.

- History Flow. Ejem.(Mostrar flujo de información de uno de los autores de la última versión).  
*<http://localhost/wiki/es.wikipedia.org/UCA/individual/Emijrp>*
- Prefuse <sup>3</sup>. Ejem. (Mostrar gráfico artículo wiki-UCA, en formato Prefuse). *<http://localhost/wiki/prefuse/es.wikipedia.org/UCA/>*
- Gephi <sup>4</sup>.. Ejem. (Mostrar gráfico del artículo wiki-UCA, en formato Gephi). *<http://localhost/wiki/gephi/es.wikipedia.org/UCA/gml>*

Todos estos tipo de medios se suministran a partir de medios estándar predefinidos en la API de Restlet en la clase *org.restlet.data.MediaType*, [6] , incluyendo la posibilidad de definir nuestros propios *media types* si necesitásemos un tipo de medio que se ajustase a unas características diferentes.

---

<sup>2</sup>Término usado para definir una Subversión la cual especifica sucesivas revisiones de versiones de uno o varios documentos.

<sup>3</sup>Conjunto de herramientas software para la visualización gráfica de datos en modo interactivo, proporcionando soporte para su programación en Java.

<sup>4</sup>Herramienta software para la visualización interactiva y la exploración de datos con la cual ayuda al usuario la posibilidad establecer determinadas hipótesis

Nombre	Descripción
text/plain	texto plano
text/html	documento html
application/xml	documento xml
application/json	documento json
image/jpeg	imagen en formato jpeg
image/png	imagen en formato png
application/pdf	documento pdf

Tabla 1.1: Algunos Tipos de Medios estándar soportados por Restlet

Como puede verse en la *Tabla 1.1* represento algunos de los tipos de medios estándar disponibles en dicha clase. Para este proyecto utilizaremos algunos de ellos como las salidas en png, xml <sup>5</sup>, html, texto plano.

## 1.2. Motivación

La principal motivación a la hora de preparar el Proyecto Fin de Carrera fue el poder aplicar otro lenguaje de programación que me permitiera realizar operaciones a nivel web lo mas flexible posible y que estuviera relacionado con la puesta en marcha de algún tipo de aplicación web.

De entre varios lenguajes de programación que estuviesen orientados al entorno web (PHP, Java, ASP.NET ...), decidí que para tal fin usaría *Java*. Me pareció bastante interesante tanto desde el punto de vista laboral como desde el punto de vista personal.

## 1.3. Alcance

Para conseguir que la aplicación, como *servicio web*, sea operativa, se crea un envoltorio que recoja el funcionamiento de la aplicación History Flow proporcionándole los datos necesarios para su visualización On-Line. La aplicación de escritorio se apoya originariamente en plugins que son los encargados de proporcionar la información necesaria.

El alcance del proyecto lo podemos ramificar en función de dos aspectos:

- Orígenes de los datos.
- Motores de visualización gráfica.

---

<sup>5</sup>Lenguaje de marcas extensible(XML), con formato muy simple y texto flexible derivado de SGML (ISO 8879).

### 1.3.1. Orígenes de los datos

Datos obtenidos a partir de plugins utilizados en el *Servicio Web*, WebHistoryFlow, para el cálculo de nuevos resultados, como:

- Plugin para archivos Wikimedia (wiki) : Importa datos XML de los distintos servidores en los cuales están alojados ficheros con información en formato wiki, como puede ser *es.wikipedia.org*. Este plugin fue desarrollado por Fernanda V. Biegas [16] , Martín Wattenberg [17] y Martín Wattenberg [18].
- Plugin para archivo perteneciente a una Subversión (svn) : Importa información existente en los servidores que gestionan las distintas versiones de uno o mas ficheros, clasificándolos en Subversiones, como puede ser *svn.apache.org*. El plugin fue desarrollado por Richard Feard.

### 1.3.2. Motores de visualización gráfica

Motores a través de los cuales mostraremos distintos tipos de representaciones gráficas en función de los datos obtenidos. Estos motores de visualización gráfica estarían formados por:

- La visualización gráfica con formato History Flow, consiguiendo una transformación del formato original en un nuevo apto para su visualización en entornos online.
- La representación gráfica a partir del framework de Prefuse, aplicándole a una demo previamente ajustada, los nuevos datos dinámicos que son generados en formato *xml* de Prefuse.
- Gráficos realizados con el framework de Gephi. Interpretando los datos obtenidos de wikis y ficheros de svn para su transformación en información gráfica con formato Gephi, una vez calculado el correspondiente fichero con formato *gml* de Gephi.

Sin embargo aunque el alcance del proyecto es el anteriormente expuesto, la solución propuesta es flexible, extensible y suficientemente comentada como para que en futuros trabajos se pueda ampliar con otros orígenes de datos y motores de visualización gráfica.

## 1.4. Metodología de desarrollo

Con el fin de poder llegar a comprender como funcionaba Java y analizar lo que se podía hacer, decidí empezar desde la base y avanzar poco a poco, recopilando el máximo de información sobre muchos de los puntos claves de Java, independientemente de que lo aplicase o no en el proyecto (Tipos de datos, estructuras de datos, applets, servlets, JSP, hilos, eventos, menús, formularios).

Una vez revisados todos estos aspectos y mirado las concordancias existentes entre Java y C++ pasaría a analizar el software de la aplicación History Flow que está realizado con Java. Este proceso de estudio del lenguaje Java, que ha durado aproximadamente dos meses (octubre-noviembre 2010), no ha sido incluido en este proyecto, ni como tiempo para la realización del mismo ni como coste añadido para su realización, ya que lo he considerado como un período de formación personal.

La metodología de desarrollo a usar en el proyecto definirá el conjunto de pasos y procedimientos a seguir para desarrollar el producto software. El proyecto utilizará una metodología que seguirá un *Modelo Lineal Secuencial*[\[12\]](#), como se especifica en el modelo tipo de la *figura 1.1*.

Se usará un *modelo lineal secuencial* ya que la forma y el modo en la realización de éste proyecto se ajusta a las operaciones definidas en este tipo de modelos: análisis de requisitos del sistema, análisis del software, diseño, codificación, pruebas y mantenimiento.

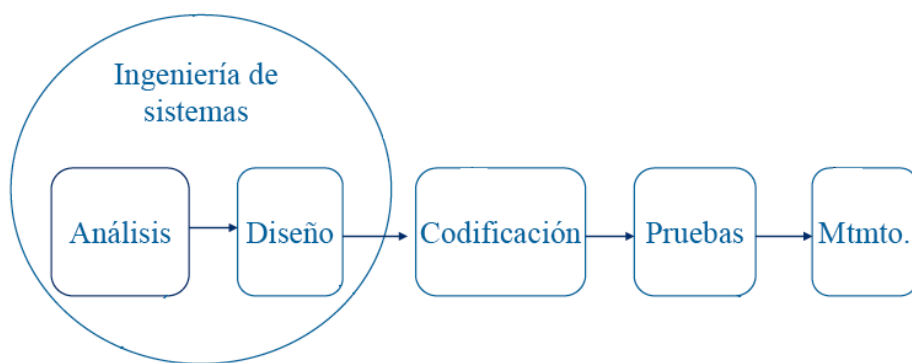


Figura 1.1: Metodología de desarrollo de un Modelo Lineal Secuencial.

Dividiremos el proceso de desarrollo en un conjunto de etapas o fases secuenciales las cuales estarán delimitadas concretamente por la definición de *hitos* y hasta que no concluyan éstos no pasará al siguiente. Como es común en este tipo de modelo de desarrollo, solo se realizarán iteraciones en fases, actividades o tareas contiguas.

A continuación describo algunas de las razones por las que utilizaré este método de desarrollo :

- Habrá que ir acabando fases para poder avanzar a la siguiente
- Me permitirá establecer unas fechas de comienzo y finalización concretas, imprescindibles para el proyecto fin de carrera.
- Y como es obligado en este tipo de método de desarrollo tendré que revisar el progreso del proyecto a medida que se va realizando.
- Evitaremos el llamado *estados de bloqueo*, característico de este tipo de modelo. Lógicamente no se producirá ya que solo existe una persona y no habrá que esperar a la finalización de determinadas tareas por parte de otros miembros del equipo.

## 1.5. Planificación

Con el fin de especificar de forma clara el orden de las tareas y estimar el tiempo necesario para la realización del proyecto se describen los siguientes apartados que se citan a continuación.

### 1.5.1. Objetivo inicial

El objetivo inicial y en modo generalizado de este proyecto era modificar una aplicación de escritorio, encapsularla, y teniendo en cuenta unos determinados parámetros aportados por el cliente, proporcionar un gráfico.

A medida que el proyecto ha ido avanzando, este objetivo ha evolucionado hacia un enfoque un poco mas ambicioso como es la creación en un *Servicio Web*.

### 1.5.2. División del trabajo

La división y descomposición del proyecto se realizará a partir de los objetivos especificados en la raíz del árbol, así mismo, se indica el trabajo de desarrollo a realizar en los distintos subniveles de los principales objetivos.



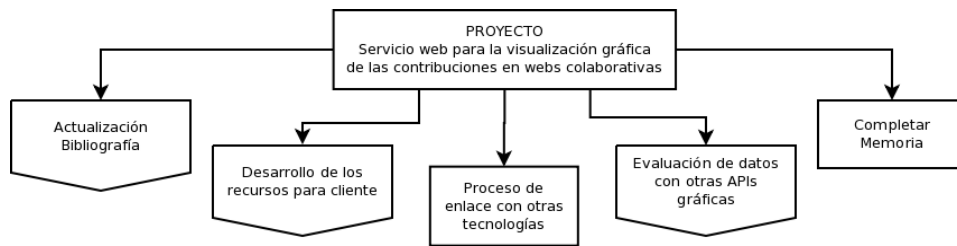


Figura 1.2: Estructura desglosada descomposición trabajo.

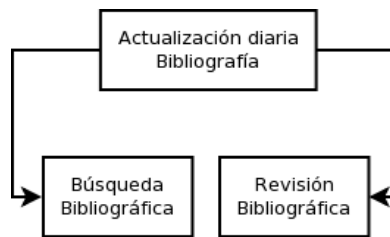


Figura 1.3: Módulo Bibliografía.

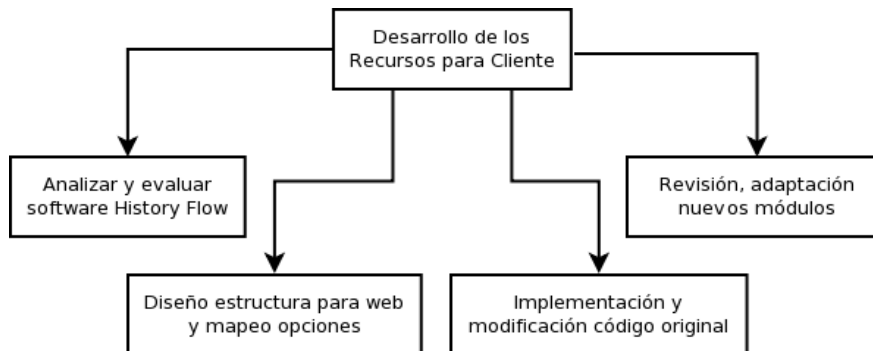


Figura 1.4: Módulo Recursos.

### 1.5.3. Estimación de tiempos

Con relación a los objetivos anteriormente especificados y teniendo en cuenta la disposición personal para la realización del proyecto, especifico la estimación de tiempos previsto. Lógicamente todos los tiempos estarán expuestos a posibles imprevistos.

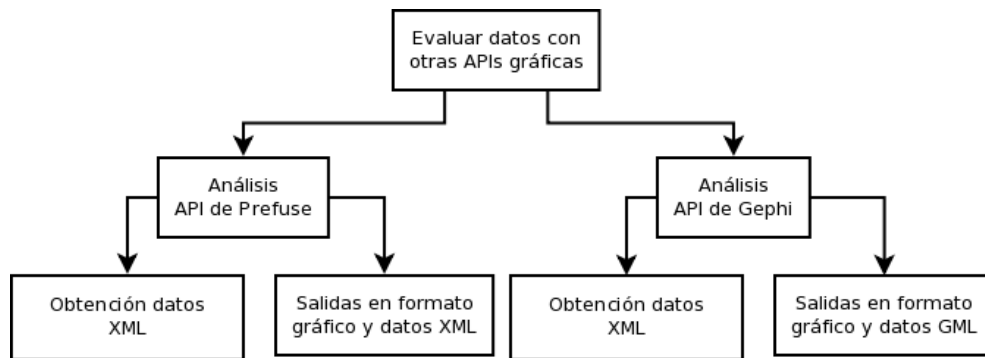


Figura 1.5: Módulo APIs externas.

ACTIVIDAD	DURACIÓN ESTIMADA
Búsqueda bibliográfica	8 semanas
Revisión bibliográfica	2 semanas
Análisis software History Flow	3 semanas
Diseño estructura y mapeo recursos	2 semanas
Implementación, modificación código fuente	5 semanas
Revisión, adaptación otras tecnologías	3 semanas
Evaluación, análisis API de Prefuse	1 semanas
Obtener salidas en modo texto y gráfico para Prefuse	1 semanas
Evaluación, análisis API de Gephi	1 semanas
Obtener salidas en modo texto y gráfico para Gephi	1 semanas
Completar memoria	5 semanas
Esfuerzo total	32 semanas (aproximadamente 8 meses)

Figura 1.6: Estimación de tiempos del proyecto.

#### 1.5.4. Identificación de hitos

Para poder tener una idea de cómo se va desarrollando el proyecto y si se van cumpliendo los objetivos, estableceré unos hitos estableciendo unos objetivos intermedios. En buena lógica, la realización en tiempo o la obtención de los resultados deseados de los hitos establecidos, conducirán a la finalización del proyecto.

- H1 : Finalización investigación bibliográfica.
- H2 : Envío recursos al cliente en modo WebHistoryflow.
- H3 : Envío recurso al cliente con opciones API Prefuse.
- H4 : Envío recurso al cliente con opciones API Gephi.
- H5 : Consecución objetivo final.

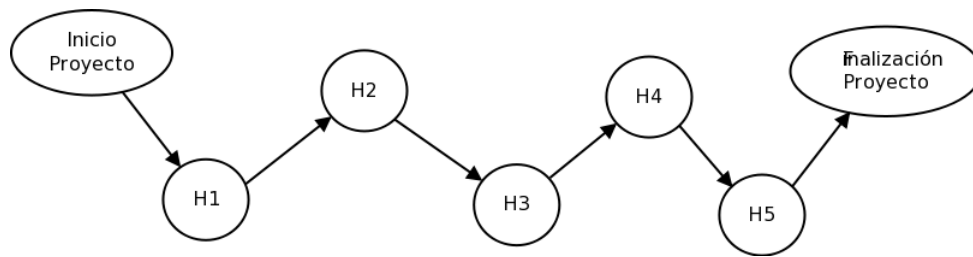


Figura 1.7: Hitos para la consecución del objetivo final.

### 1.5.5. Encadenamiento de actividades

A continuación relaciono el conjunto de tareas previstas, determinando la relación entre las distintas tareas para realizar el proyecto en función de los objetivos y tiempos previstos para su realización. a través de los nodos indico las tareas que intentaré realizar a medida que avanza el proyecto y con las flechas el orden en que se prevean realizar.

Cada tarea incluye en su parte superior dos anotaciones, la fecha prevista de comienzo (*superior izquierda*) y otra con las semanas que se invertirá en su realización (*superior derecha*).

Durante el proceso de desarrollo se indican también los hitos anteriormente definidos. Éstos están posicionados en la red de actividades con fechas previstas de conclusión de objetivos.

A pesar de que algunas de las tareas indicadas en el diagrama de la *figura* 1.8 se podrán realizar en paralelo, ya sea por necesidad en la implementación o por no caer en el aburrimiento, no se muestran en éste diagrama y serán especificados en el *diagrama de Gantt*[19] de la *figura* 1.9.

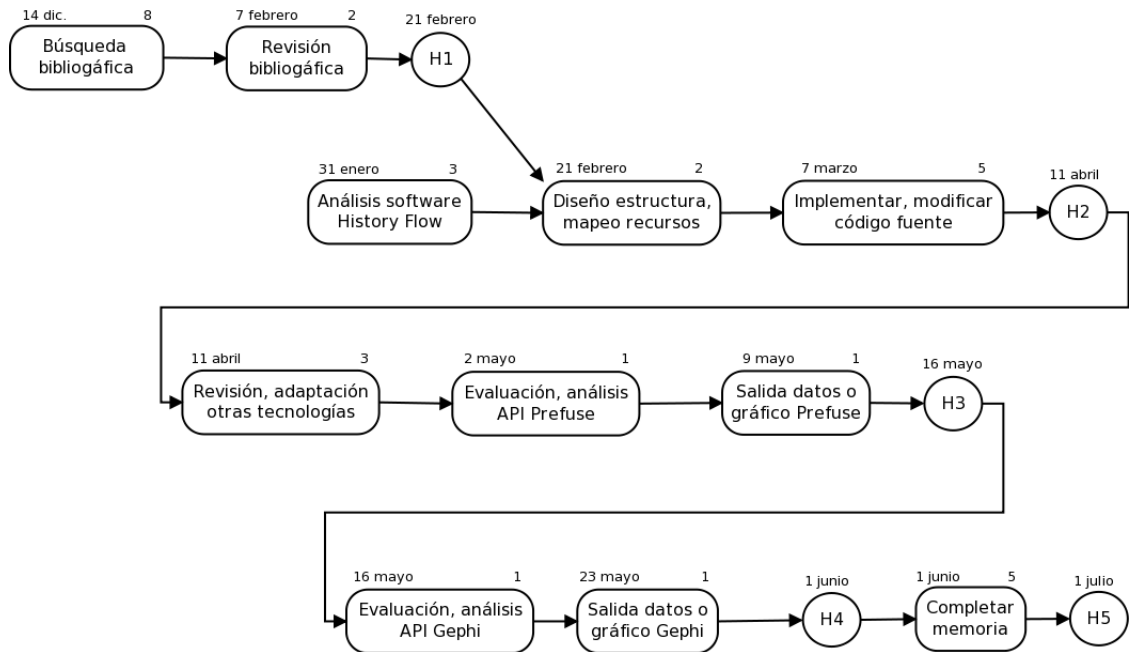


Figura 1.8: Red de actividades previstas.

### 1.5.6. Planificación temporal

La planificación temporal se ha preparado con el software OpenProj y utilizando su opción para la realización de diagramas de gantt. A través de este diagrama de gantt[19] se establecen la asignación temporal prevista para la realización de las actividades, indicando explícitamente los tiempos e indicando aquellas tareas que pueden realizarse a lo largo del proyecto.

Los rombos indicados en la *figura* 1.9 de color negro representan la relación de hitos anteriormente especificados en la *figura* 1.7.

La tareas especificadas como simultáneas se han de entender como posibles cambios de tarea a lo largo de un día, en el diagrama se ha indicado con barras en color azul, indicando su posibilidad de desplazamiento, en distintos puntos del proyecto, a lo largo de la vida de éste.

Los trazos indicados en la *figura* 1.9 en color rojo, indican tareas críticas, las cuales deben de realizarse en el orden y la posición indicados. Lo especificado en color azul son tareas que pueden realizarse o desplazarse a lo largo del proyecto sin que ello suponga un retraso en la realización del mismo.

Tarea o actividad	Días	Fecha-Hora Inicio	Fecha-Hora Fin
Búsqueda bibliográfica	40	14/12/10 08:00	07/02/11 17:00
Revisión bibliográfica	10	08/02/11 09:00	22/02/11 09:00
Hito 1	0	22/02/11 09:00	22/02/11 09:00
Análisis software History Flow	15	31/01/11 08:00	18/02/11 17:00
Diseño estructura, mapeo recursos	10	23/02/11 08:00	08/03/11 17:00
Implementación, modif. History Flow	25	09/03/11 08:00	12/04/11 17:00
Hito 2	0	12/04/11 17:00	12/04/11 17:00
Revisión, adaptación otras tecnologías	15	13/04/11 08:00	03/05/11 17:00
Evaluación, análisis API Prefuse	5	04/05/11 08:00	10/05/11 17:00
Representación recurso API Prefuse	5	11/05/11 08:00	17/05/11 17:00
Hito 3	0	17/05/11 17:00	17/05/11 17:00
Evaluación, análisis API Gephi	5	18/05/11 08:00	24/05/11 17:00
Representación recurso API Gephi	5	25/05/11 08:00	31/05/11 17:00
Hito 4	0	31/05/11 17:00	31/05/11 17:00
Completar memoria	25	01/06/11 08:00	05/07/11 17:00
Hito 5	0	05/07/11 17:00	05/07/11 17:00
<b>Total días</b>	<b>160</b>		

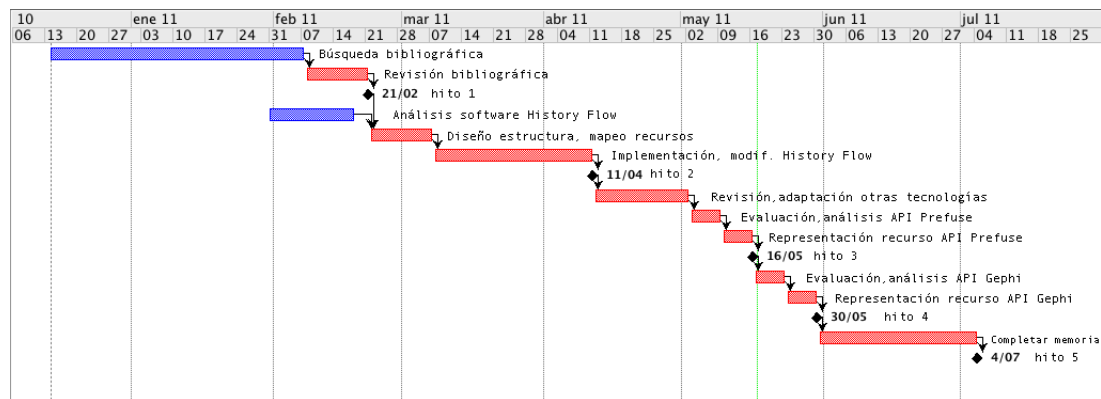
Tabla 1.2: Relación actividades Diagrama de Gantt para la *figura 1.9*.

Figura 1.9: Diagrama de Gantt PFC - relación temporal actividades WebHistoryFlow.

## 1.6. Coste de desarrollo

La estimación del coste de desarrollo se establecerá en función de un calendario base estándar con unos parámetros predefinidos para poder obtener una estimación del coste y las horas invertidas, sin llegar a especificar al detalle las horas invertidas cada día y cada momento del día, ya que eso conllevaría un control muy exhaustivo de las horas aplicadas.

<b>Tarea o actividad</b>	<b>Días</b>	<b>Horas</b>	<b>Coste €</b>
Búsqueda bibliográfica	40	320	7680
Revisión bibliográfica	10	80	1920
Hito 1	0	0	0
Análisis software History Flow	15	120	2880
Diseño estructura, mapeo recursos	10	80	1920
Implementación, modif. History Flow	25	200	4800
Hito 2	0	0	0
Revisión, adaptación otras tecnologías	15	120	2880
Evaluación, análisis API Prefuse	5	40	960
Representación recurso Prefuse: datos, gráfico	5	40	960
Hito 3	0	0	0
Evaluación, análisis API Gephi	5	40	960
Representación recurso Gephi: datos, gráfico	5	40	960
Hito 4	0	0	0
Completar memoria	25	200	4800
Hito 5	0	0	0
<b>Totales</b>	<b>160</b>	<b>1280</b>	<b>30720</b>

Tabla 1.3: Relación de costes para el desarrollo del proyecto.

Se tendrá en cuenta que los parámetros para su evaluación serán los siguientes a partir de un calendario estándar :

- Personas implicadas en proyecto : 1
- Horas por día : 8 h.
- Importe por hora : 24 €
- Días a la semana : 5

En la planificación detallada de la *tabla* 1.3, no están incluidas todas las posibles incidencias que puedan ocurrir a lo largo del proyecto como pueden ser: enfermedad, festivos, imprevistos etc..., así mismo tampoco están incluidos los tiempos empleados para la formación personal antes de la realización del proyecto.

## Capítulo 2

# Tecnologías usadas

Para la realización del proyecto he utilizado varias herramientas con la distinción GNU o de libre utilización. Todas han sido seleccionadas con la intención de conseguir los máximos resultados y sobre todo que fuesen fáciles de usar.

A medida que fue desarrollándose el proyecto y con la ayuda del tutor he ido seleccionando de, entre varias, las que mejor se adaptaba a mis necesidades desde el punto de vista del desarrollo de las distintas partes que engloba el proyecto.

Podríamos clasificarlas en dos apartados:

- Entornos y herramientas de desarrollo.
- Lenguajes y frameworks de desarrollo software.

### 2.1. Entornos y herramientas de desarrollo

A la hora de realizar cualquier actividad en el proyecto la idea era no perder tiempo con aplicaciones complicadas, obteniendo resultados simples para centrarme en el desarrollo del proyecto.

Para poder llevar un cierto orden en la generación del proyecto y con la finalidad de ir agrupando toda la información que encontraba, he usado las aplicaciones que a continuación relaciono. Todas han resultado ser bastante útiles, y como dije antes, bastante fáciles de usar.

- **GitHub.** A pesar de haber visto un software similar a GitHub, que es Assembla, decidí por motivos a mi entender de facilidad, la utilización del software Github. GitHub es un software para alojar los proyectos

utilizando un sistema de control de versiones <sup>1</sup> Git <sup>2</sup> . Desarrollado por GitHub Inc. y programado con el framework Ruby on Rails.

- **RapidSvn.** Una forma cómoda de gestionar todo lo que se va incluyendo en GitHub es usando este gestor de software para alojar proyectos. De una forma muy cómoda puedes realizar las subidas, bajadas y modificaciones de todas las versiones que vas incluyendo en el servidor donde tienes alojado el proyecto. Es un software *open source* y puedes bajarlo desde aquí <http://rapidsvn.tigris.org/> con un sistema de subversiones realizadas en C++
- **IDE Netbeans.** El análisis, modificación y adaptación web del software History Flow se ha realizado con el IDE Netbeans aunque igualmente se podría haber realizado con el IDE Eclipse. La opción con Netbeans ha sido por su facilidad visual a la hora de elegir las distintas opciones. Descansa sobre una implementación Java, aunque es notable la velocidad de ejecución de Eclipse.
- **JabRef.** Es un software open source que ayuda a mantener una relación bibliográfica de todos los enlaces, archivos, referencias etc..., que se van recopilando a lo largo del proceso de investigación. Corre bajo una VM de Java y es de gran utilidad a la hora de reconstruir y localizar la información recopilada. Localizado en <http://jabref.sourceforge.net/>
- **TexMaker.** Software muy sencillo de usar para la realización de trabajos en Latex <sup>3</sup> y está liberado bajo licencia GPL. Dispone de menús e iconos que hacen muy intuitivo la implementación y escritura de trabajos en Latex y está disponible para Linux, MacOSx y Windows.
- **Dia ver. 0.97.1** Es un *Editor de Diagramas*, es software libre GNU con Licencia General Pública (GPL), para desarrollar gráficos relacionados con distintos tipos de plantillas para representaciones como:
  - Diagramas UML
  - Cisco Computer
  - Map Isométrico
  - Diagramas de lógica
  - NetWork

---

<sup>1</sup>Control de versiones es la gestión de los cambios que se realizan sobre los elementos de un producto, como una wiki, una svn etc... facilitando la administración de las mismas.

<sup>2</sup>Software de control de versiones diseñado por Linus Torvalds sobre todo cuando se trata de una gran cantidad de archivos.

<sup>3</sup>Sistema de composición de textos, muy utilizado para representación de libros, proyectos, documentos técnicos y científicos.



Estos solo son algunas de las plantillas que se pueden utilizar. Por supuesto, ni que decir queda, que, su utilización es muy fácil, flexible e intuitiva, ideal para no perder el tiempo y obtener buenos resultados. Sus datos son exportables a diversos formatos como el formato gráfico *.png*, ideal para incluir en proyectos.

- **OpenProj.** En mi caso haremos uso del software OpenProj para el desarrollo del Gráfico Gantt del proyecto. El desarrollo del gráfico gantt de éste proyecto está representado en función de una persona y por ello no se mostrarán una relación de actividades solapadas ya que no será posible realizar simultáneamente dos tareas al mismo tiempo.

La única forma en la que se podrían expresar a mi entender solapamiento, habiendo realizado el proyecto una persona, sería en la realización alterna de dos o mas actividades, pero aun habiendo realizado dichos solapamientos en pequeños trozos de actividades durante el proyecto, no las he descrito en el diagrama para poder visualizar una representación mas clara y no ensuciar su representación. Aún así, daría pie a confusiones a la hora de establecer el tiempo total de ejecución del proyecto, con lo cual todas las actividades representadas estarán en modo secuencial.

OpenProj puede ser utilizado bajo licencia CPAL (Common Public Attribution License Versión 1.0) y es un software que proporciona la posibilidad de generar, entre otros, gráficos tipo:

- Diagramas Gantt.
- Diagramas de redes (PERT).
- Diagramas WBS.
- Diagramas para Caracteres RBS.

## 2.2. Modelado orientado a objetos

Para realizar el análisis de este proyecto utilizaremos una *metodología orientada a objetos* [12] ya que la justificación del uso de dicha metodología se ajusta a características como:

- La orientación a objetos se aproxima más a la forma de pensar de las personas. Esto lo hace más comprensible y fácil de aplicar.
- Mejora la intercomunicación entre expertos en el dominio del problema, usuarios, analistas y diseñadores.
- Proporciona abstracción, ya que en cada momento se consideran sólo los elementos que nos interesan descartando los demás.

- Aumenta la consistencia interna al tratar los atributos y las operaciones como un todo.
- Permite expresar características comunes sin repetir la información.
- Facilita la reutilización de diseños y códigos.
- Facilita la revisión y modificación de los sistemas desarrollados.
- Origina sistemas más estables y robustos.

## 2.3. Lenguajes y frameworks de desarrollo software

Para la implementación de los distintos apartados del proyecto he necesitado tirar de distintas tecnologías ya que el resultado a mostrar exigía el cambio en el tipo de recurso a suministrar. La mayor parte del software usado es a través de implementaciones en Java.

### 2.3.1. Prefuse

Es una herramienta software para la visualización de datos de modo interactivo. Existe un framework de Prefuse *toolkit* utilizado para la programación en Java. Gestiona datos en varios formatos, entre ellos el formato *xml* usado en este proyecto. Está licenciado bajo términos de una *licencia BSD*, para su uso libre ya sea comercial o no.

El kit de herramientas de Prefuse ofrece las siguientes características:

- Estructuras de datos de E/S para bibliotecas gráficas no estructuradas y datos en formato árbol.
- Múltiples visualizaciones a partir de un único origen de datos.
- Múltiples puntos de vista de una visualización única.
- Escalabilidad de miles de elementos en pantalla, y el apoyo de fuentes de datos con millones de elementos.
- Procesamiento por lotes de los datos.
- Una biblioteca para disposiciones previstas y técnicas de distorsión predefinidas.
- La animación y el tiempo de procesamiento basado en la transformación de gráficos, como la panorámica y zoom.
- Un simulador de la fuerza física para el diseño y la interacción.
- Mapas integrados a todo color y con funcionalidades de búsqueda.

- Registro de eventos para apoyar la evaluación de visualizaciones.

En la *Figura 2.1* se muestra la representación de la arquitectura, las abstracciones base y las bibliotecas para el proceso de visualización de la información. Prefuse a partir de las listas de acciones filtra los datos abstractos en contenido visualizable y les asigna las propiedades visuales (posición, color, tamaño, tipode letra, etc...).

Los datos se procesan por módulos, siempre en función de cada elemento a través de un *RendererFactory* <sup>4</sup>, apuntando a *Items Visuales* para la construcción de pantallas interactivas, con lo que en cualquier momento un usuario puede realizar cambios en el framework de forma interactiva.

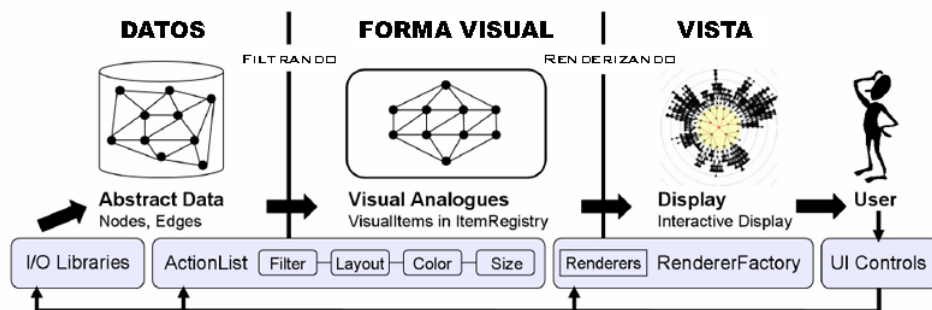


Figura 2.1: Estructura de visualización del framework de Prefuse.

### 2.3.2. Gephi

Es un proyecto *Open Source* <sup>5</sup> desde el cual se pueden realizar análisis de datos en función de representaciones gráficas. Gephi además de una aplicación de escritorio muy potente en cuanto a variedad de visualizaciones y facilidad de uso, dispone de un framework para la programación en Java.

De entre las muchas características que posee Gephi [23] a continuación relaciono algunas de ellas :

- Visualización Real-time. Impulsado por el motor ad-hoc OpenGL <sup>6</sup>, Gephi realiza exploraciones de la red de manera interactiva y eficiente.
- Iteración de la visualización mediante filtrado dinámico.

<sup>4</sup>Clase pública abstracta de la API de Prefuse encargada de proporcionar una instancia de renderización adecuada para dibujar un *Item Visual*

<sup>5</sup>Código abierto, se refiere al software que se distribuye y desarrolla libremente.

<sup>6</sup>Open Graphics Library, especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D

- Amplia variedad de herramientas para la manipulación gráfica.
- Diseño. Gephi permite al usuario cambiar la configuración del diseño durante la ejecución.
- Métricas. Ofrece estadísticas e indicadores como los observados en la *Figura 2.2* para las métricas más comunes para el análisis de redes sociales (ARS) y las redes scale-free. Scale-free es una red cuyo grado de distribución sigue una ley de potencia, por lo menos asintóticamente. Es decir, la fracción  $P(k)$  de nodos de la red que tiene conexiones con otros nodos  $k$  va hacia valores grandes de  $k$  como

$$P(k) \sim ck^{-\gamma}$$

donde  $c$  es una constante de normalización e  $\gamma$  es un parámetro cuyo valor está típicamente en el rango  $2 < \gamma < 3$ , aunque en ocasiones puede estar fuera de estos límites. Por ejemplo World Wide Web, las redes de citación, redes biológicas, las redes de la compañía aérea y algunas redes sociales.

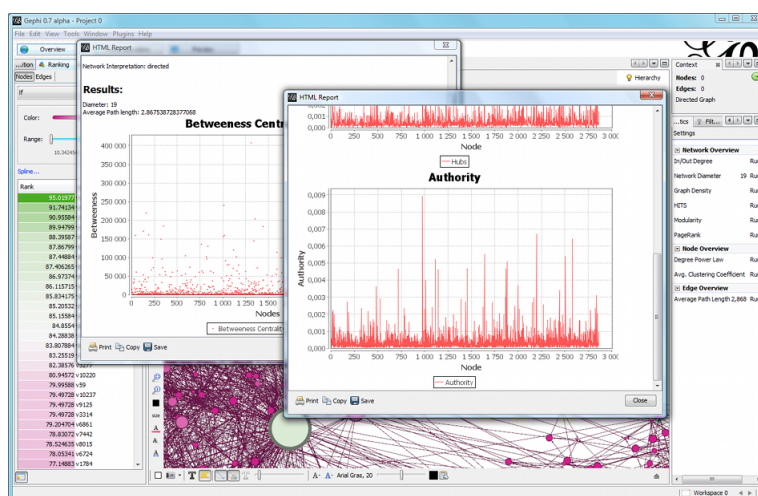


Figura 2.2: Aplicación de escritorio de Gephi.

- Análisis dinámico de redes. Gephi es la plataforma ideal para el análisis de redes dinámicas (ADN). Estructuras dinámicas, como las redes sociales se pueden filtrar con el componente de *línea de tiempo*.
- Creación cartográfica. Utiliza datos de clasificación o partición para hacer significativa la representación de la red, personalizar los colores, el tamaño o las etiquetas para poner sentido a la representación de la red.

- Filtrado dinámico. Selecciona los nodos y/o las aristas sobre la base de la estructura de red o datos utilizando la interfaz de usuario interactiva para filtrar la red en tiempo real.
- Plugin center. El built-in Plugins Center obtiene automáticamente una lista de plugins disponibles en el portal de Gephi Plugin y se encarga de todas las actualizaciones de software.

### 2.3.3. History Flow

History Flow representa cada documento como una línea vertical cuya longitud se corresponde con la longitud del documento. A continuación mostraré un ejemplo de como realizaría dicho trabajo.

Pongamos por ejemplo, tres personas que hagan contribuciones a una página wiki en diferentes puntos en el tiempo. Cada persona que edita la página y luego la guarda, sus cambios, hacen que se convierta en la versión mas reciente de esa página.

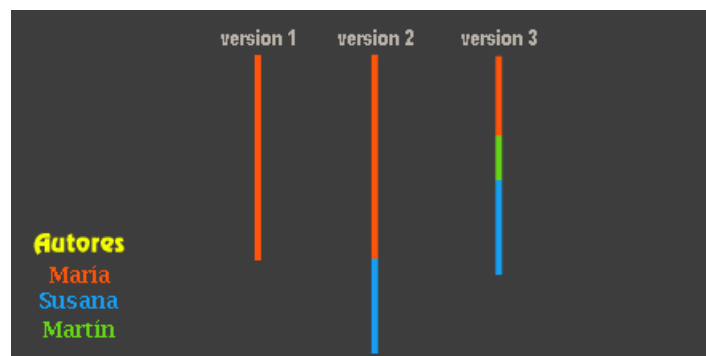


Figura 2.3: Creación primera versión en una wiki, *ver.1*

La línea roja vertical de la *Figura 2.3* representa la primera versión del documento wiki y puesto que María es la que crea todos los contenidos de esta página solo se refleja su color. Como dije antes la línea indica la cantidad de texto que María ha escrito.

En la *Figura 2.4* Susana añade un trozo de texto al final de la entrada del original de María, fíjate como la línea azul de Susana se anexa al final de la línea roja de María, indicando que el texto de susana se añadió al final de la página. Susana guarda los cambios y ésta versión se convierte en la última versión de la página.

En la *Figura 2.5* Martín le parece el texto de la última versión añadida demasiado detallado y lo que hace es eliminar algunas partes del texto y escribir su propia versión mas corta entre el texto de introducción de María y la última contribución de Susana.

Figura 2.4: Creación y modificación de la ultima versión, *ver.2*Figura 2.5: Eliminación y modificación de la última versión, *ver.3*Figura 2.6: Inserción en la última versión, *ver.4*

En la *Figura 2.6* referida a la versión 4, Susana realiza una pequeña contribución entre medio del texto que escribió María que aún permanece en el flujo de información desde las versiones anteriores. Como se indicó anteriormente ésta inserción se visualiza en la longitud proporcional al texto incluido.

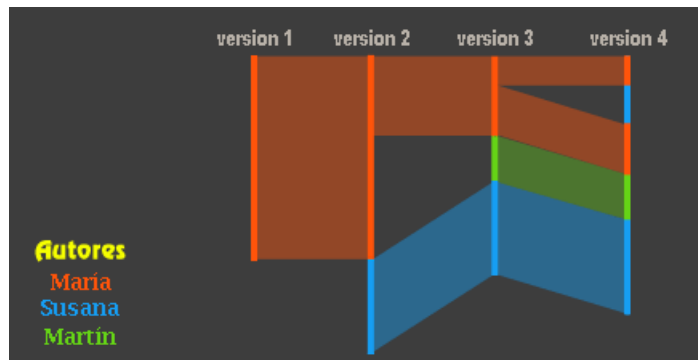


Figura 2.7: Conexión de versiones a través de paralelogramos.

Ahora una vez solicitamos a History Flow que muestre el flujo de la historia de ésta wiki, mostrará la representación indicada en la *Figura 2.7*, conectando las franjas de línea que se han mantenido respecto de un mismo autor a través de las sucesivas versiones.

Cuando no hay correspondencia en el flujo entre la versión anterior o la siguiente, se visualiza una *brecha* en su visualización que indica que en ese punto se ha realizado una inserción o un eliminación de texto.

#### 2.3.4. Modos de visualización

Para establecer distintas conclusiones de la visualización que muestra History Flow, se proporciona cinco modos distintos de representación del gráfico. Los posibles tipos de visualización son los siguientes:

- **Grupo** : Es el modo que tiene activado por defecto, se visualizan las contribuciones de los distintos autores a lo largo de la vida de ese artículo. Cada autor estará representado por un color determinado. *Figura 2.8*.

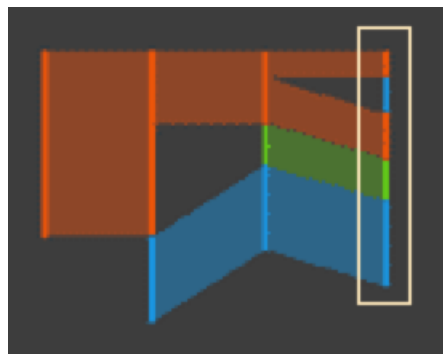


Figura 2.8: Representación de versiones en modo grupo.

- **Autor individual** : Este tipo de representación visualiza en modo resaltado el flujo de la(s) aportaciones de un autor determinado a lo largo de las distintas contribuciones o versiones. También resaltará el autor que se está procesando con respecto a los demás para que se vea claramente a quien representa el flujo resaltado que se representa en el gráfico. *Figura 2.9.*

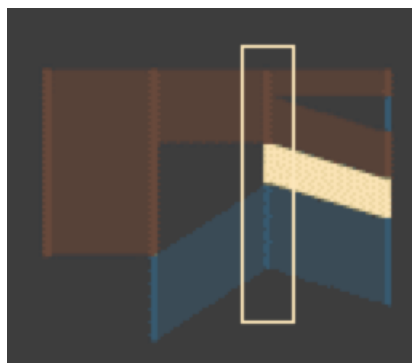


Figura 2.9: Representación de versiones en modo autor individual.

- **Antigüedad** : En este tipo de visualización gráfica todos los flujos de los distintos autores se representan a través de tonos degradados de grises representando la persistencia de las diferentes contribuciones. Las tonalidades van desde un blanco claro, representando contribuciones recientes, hasta un gris oscuro, que interpretan contribuciones antiguas. *Figura 2.10.*

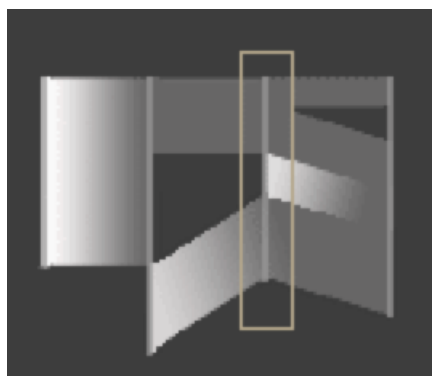


Figura 2.10: Representación de versiones en modo antigüedad del flujo de datos.

- **Cambios recientes** : Con este modo de visualización se muestra cual es el contenido nuevo aportado en cada versión de ese artículo, independientemente de quien lo haya realizado. Con esta visualización podemos



### 2.3. LENGUAJES Y FRAMEWORKS DE DESARROLLO SOFTWARE 23

ver qué porciones del texto han sido mas o menos modificados a lo largo del tiempo. *Figura 2.11.*

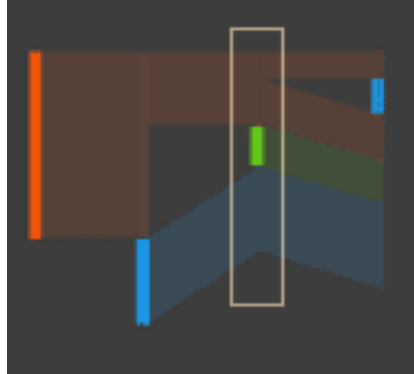


Figura 2.11: Representación de versiones en modo cambios recientes en cada versión.

- **Links** : Con este tipo se muestra la colocación de los links de comienzo de una página, representado por un cuadrado y cual es la persistencia de éste a lo largo del flujo de vida del artículo. *Figura 2.12.*

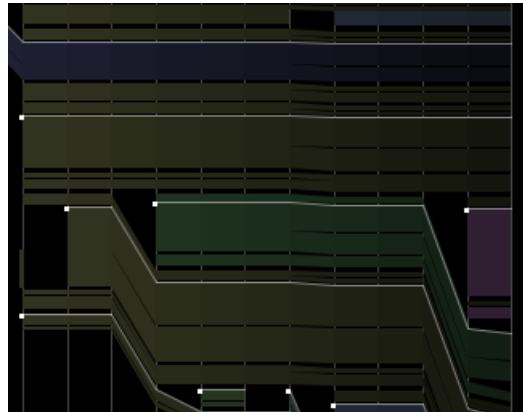


Figura 2.12: Representación de versiones en modo links de comienzo en cada versión.

#### 2.3.5. Restlet

Será a partir de este framework, *Restlet* [5], desde el cual se va a permitir proporcionar un formato de URI's sencillo para poder identificar de manera inequívoca la identificación de un recurso. Por ejemplo

`http://localhost/wiki/org.wikipedia.es/UCA` , donde se está solicitando un recurso, *wiki*, para que se procese un artículo de la wikipedia de nombre *UCA*.

Restlet es un *framework* creado por Jerome Louvel (Noelios Technologies), es un software libre desarrollado en Java y realiza un sencillo mapeo de los principios REST. Este framework es el que he utilizado como base para realizar todo el mapeo de los recursos solicitados por el cliente.

Restlet es un framework web de código abierto para desarrolladores de Java. Ofrece amplias capacidades REST, framework orientado a objetos y un extenso conjunto de clases y rutinas para usar o ampliar.

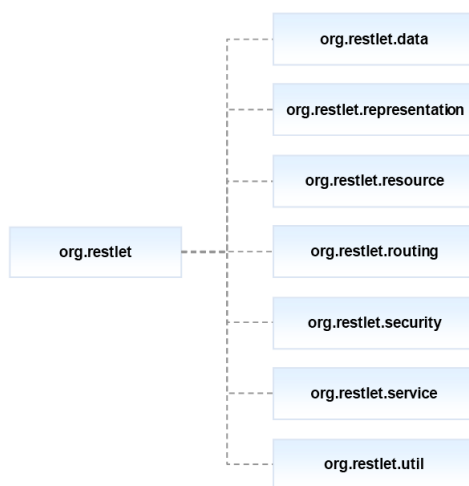


Figura 2.13: Contenido Paquete API Restlet.

El Framework Restlet puede usarse tanto para exponer como para consumir recursos de la Web. Una de las ventajas de Restlet es que proporciona un buen soporte del lado del cliente y del servidor, proporcionando todas las características de HTTP, como, métodos condicionales, negociación de contenido, compresión de contenido o su verificación.

#### 2.3.5.1. Características de Restlet

De entre las características mas relevantes de Restlet os cito alguna de ellas.

- Ofrece un mecanismo de Plantillas para realizar un enrutamiento sencillo entre los distintos elementos de nuestra aplicación. Los parámetros de las plantillas son codificados en la solicitud como atributos.

### 2.3. LENGUAJES Y FRAMEWORKS DE DESARROLLO SOFTWARE 25

- Ofrece un servicio de Tunneling que permite, entre otras cosas, que los browsers puedan emitir cualquier método HTTP encapsulándolo dentro de un método POST. Transparente de cara a la aplicación.
- Ofrece la funcionalidad completa de un servidor web.
- Provee varios conectores (Apache HTTP, HTTPS, JDBC...).
- Soporta distintas representaciones (XML, HTML, ...).
- Maneja distintos esquemas de seguridad (Basic, Digest, Amazon S3, OAuth...).
- Proporciona soporte para distintas plataformas (Java SE/EE, Google App Engine, Google Web Toolkit, Android) *Figura 2.14*.



Figura 2.14: Plataformas soportadas por el framework de Restlet.

#### 2.3.5.2. Arquitectura del Framework

Como se puede ver en la *Figura 2.15* la arquitectura del framework de Restlet [5] está desglosada de la siguiente forma.

- API Genérica (Restlet API) *Figura 2.13*
- Plugin de soporte para implementaciones alternativas (SPI).
- Implementación de Referencia (Noelios Engine)
- Integración con tecnologías externas (JDBC, JSON, gestores plantillas...)
- Aplicaciones.

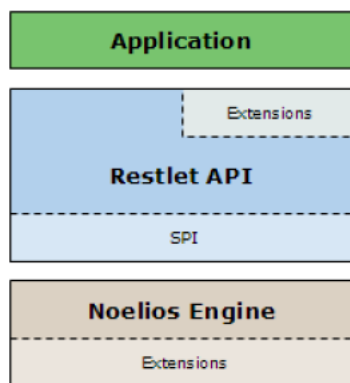


Figura 2.15: Arquitectura framework Restlet.

### 2.3.6. Java

Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los 90. La mayor parte de las tecnologías Java están bajo licencia *GNU GPL*. Sus creadores fueron James Gosling, Arthur Van Hoff y Andy Bechtolsheim.

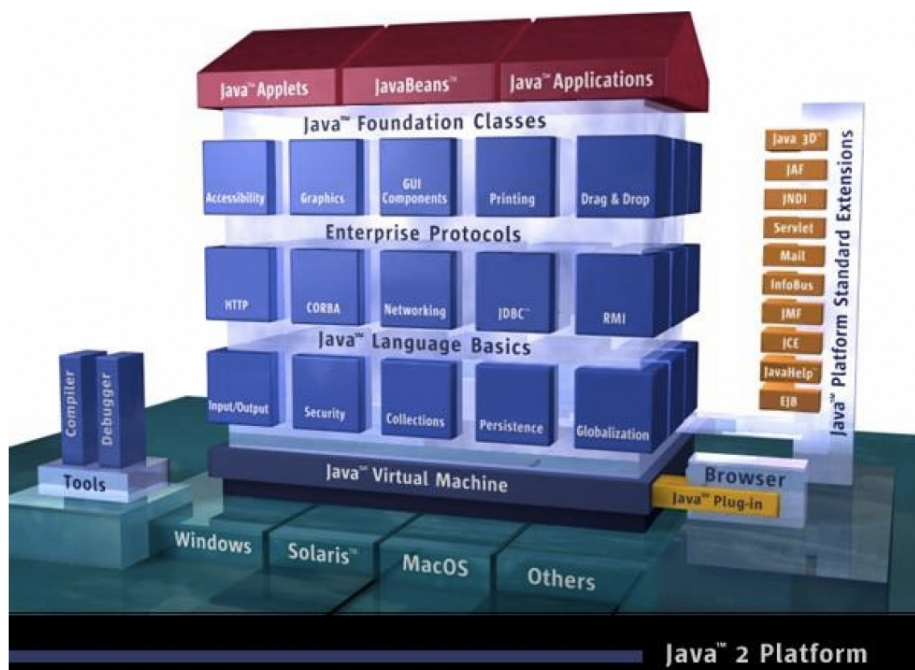


Figura 2.16: Plataforma Java 2.

Entre diciembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor

parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre. Uno de los aspectos mas interesantes de Java [11] es su característica multiplataforma como se muestra en la estructura de *Java 2* Figura 2.16.

### 2.3.7. Lenguaje UML

#### 2.3.7.1. Antecedentes UML

UML *Lenguaje de Modelado Unificado* Figura 2.17, fue creado por Grady Booch, James Rumbaugh e Ivar Jacobson. Estos señores trabajaban en empresas distintas durante la década de los años ochenta y principios de los noventa y cada uno diseñó su propia metodología para el análisis y diseño orientado a objetos. A mediados de los años noventa empezaron a intercambiar ideas entre sí y decidieron desarrollar un trabajo en común.

Los anteproyectos del UML [3] empezaron a circular en la industria del software y las reacciones resultantes trajeron consigo considerables modificaciones. Conforme diversos consorcios vieron que el *UML* era útil a sus propósitos, se creó un consorcio del UML. Entre los miembros se encuentran DEC, Hewlett-Packard, Intellicorp, Microsoft, Oracle, Texas Instruments y Rational. En 1977 el consorcio sacó la versión 1.0 del UML y lo puso a consideración del OMG (Grupo de administración de objetos) como respuesta a su propuesta para un lenguaje de modelado estándar.

El consorcio aumentó y generó la versión 1.1, la cual se puso nuevamente a consideración del OMG. El grupo adoptó esta versión a finales de 1997. El OMG se encargó de la conservación del UML y creó otras dos versiones en 1998. El UML actualmente ha llegado a ser uno de los estándares mas utilizados e importantes en la industria del software.

#### 2.3.7.2. Características UML

Para establecer qué actividades realiza este proyecto, *WebHistoryFlow*, y realizar un análisis y diseño del mismo, se utilizará el lenguaje UML ya que por sus características, relaciono algunas de ellas a continuación, proporcionará una visión clara y sin ambigüedades.

- Es un lenguaje estándar que puede utilizarse para visualizar, especificar, construir y documentar todas las actividades que componen un sistema con gran cantidad de software.
- Modela desde sistemas de información hasta aplicaciones distribuidas basadas en Web, pasando por sistemas empotrados de tiempo real.

- Es un proceso dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.
- Proporciona un vocabulario y unas reglas para utilizarlo.
- Ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos sin ninguna ambigüedad.
- Proporciona la capacidad de modelar actividades de planificación de proyectos y de sus versiones, expresar requisitos y las pruebas sobre el sistema.

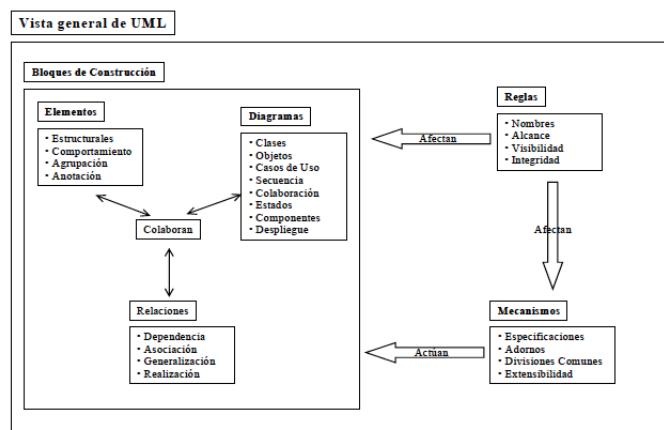


Figura 2.17: Vista general de elementos UML.

## Capítulo 3

# Especificaciones generales del proyecto

Para establecer un enfoque general de las posibilidades abarcadas tanto desde el punto de vista de ejecución del *servicio web* creado, como desde el punto de vista de restricciones impuestas en el proyecto, se describen los siguientes apartados.

### 3.1. Parámetros y funciones incluidas en WebHistoryFlow

Como el objetivo para obtener la información es realizar peticiones desde una URL, será necesario proporcionar ciertos parámetros a través de la URL, éstos, serán los que proporcionen los distintos tipos de salidas de la aplicación WebHistoryFlow.

A continuación se describe la plantilla de parámetros a indicar para realizar la petición de un determinado recurso por parte del cliente a partir de una dirección URL.

`http://spilin.uca.es:9080 / wiki | svn / [prefuse | gephi] / hostwiki | hostsvn / path / articulo | fichero / grupo | individual | reciente | tiempo | link / [usuario] / [xml | gml]`

- *wiki* / *svn* : Datos referidos a un origen tipo WikiMedia o un fichero concreto de una subversión.
- *prefuse* / *gephi* : Determina que la opción de visualización se realizará para obtener resultados en modo Prefuse o en modo Gephi (API's gráficas).
- *hostwiki* / *hostsvn* : Se refiere al host correspondiente a una wiki o a una svn.

- *path* : Camino hasta donde se encuentre la wiki o el fichero correspondiente al hostwiki o hostsvn.
- *articulo* / *fichero* : Nombre del artículo o fichero a procesar. Ya sea un tipo wiki, con lo cual hay que indicar un artículo o un tipo svn, que habría que indicar un fichero concreto de una svn.
- *grupo* / *individual* / *reciente* / *tiempo* / *link* : Modo de visualización del gráfico para la representación del flujo de información.
- *usuario* : Nombre del usuario cuando se pide una representación gráfica en función de un autor determinado, ya sea de una artículo wiki o de un fichero de una svn.
- *xml* / *gml* : Proporciona una salida en formato xml o gml para su utilización como flujo de entrada en implementación realizadas con el framework Prefuse o en la aplicación de escritorio que proporciona Gephi.

### 3.2. Especificaciones preliminares

Las primeras implementaciones de History Flow como aplicación software de escritorio, fue desarrollada en el 2003 por Fernanda B.[16] , Martín Wattenberg[17] y Kushal Dave[18].

History Flow[20] una aplicación para la visualización dinámica de la evolución de un documento a través del tiempo a partir de líneas verticales cuya longitud se va a corresponder con la longitud del documento. A estos documentos en sus distintas versiones, se les aplica el algoritmo estándar "Diff"<sup>1</sup>, utilizando la representación de paralelogramos para indicar los cambios que van surgiendo con el paso de las distintas versiones. Para identificar los distintos autores correspondientes a esas versiones, se les va asignando una clasificación por color a cada uno de ellos.

### 3.3. Restricciones generales

Debido a que se ha tratado de buscar lo esencial del software original[20], es decir, obtener un gráfico que represente el flujo de información de una wiki o un fichero de una svn, he tenido que centrarme en los aspectos y módulos que realmente interesaban y desechar aquellos apartados que aunque siendo interesantes no serán o estarán presentes en esta versión del proyecto.

Las principales restricciones del software adaptado a uso On-Line las podríamos resumir como :

---

<sup>1</sup>Utilidad para la comparación de archivos que genera las diferencias entre dos archivos o los cambios realizados en un archivo determinado comparándolo con una versión anterior del mismo archivo



- WebHistoryFlow será para uso exclusivo de ficheros con formato MediaWiki y para un fichero perteneciente a una svn.
- Las salidas gráficas están asociadas al formato establecido por History Flow y la visualización gráfica de las versiones se representarán atendiendo al espacio temporal entre cada una de ellas.
- No está contemplada la visualización de texto en función de la posición del ratón en el flujo gráfico de History Flow, sobre todo por que la intención del proyecto es obtener la visualización del gráfico y no del texto en un punto concreto de la gráfica.
- La representación gráfica en formato Prefuse se realizará solo a través de una de las demos ofrecidas en la web de Prefuse[21] en la cual se manipulan dos parámetros para establecer relaciones entre datos. Esta representación se realizará a través de un applet[22] representado a través de nodos la relación *autor-palabras*.
- La salida de datos ofrecidos en formato *xml* para el framework de Prefuse y *gml* para el framework de Gephi, solo serán en relación a dos parámetros, *autor* y *palabra*.
- Para la representación gráfica de Gephi[23] se usará un solo estilo de visualización referida a una de las demos ofrecidas en su web y la salida será formato pdf.
- Solo están contemplados un reducido numero de errores básicos ya que la mejora en aspectos como, otros errores y actualizaciones, se podrán recoger en futuras versiones.
- La visualización del applet en la opción de Prefuse se recibirá en el explorador con las proporciones de : ancho = 1400 pixels , alto = 640 pixels.

### 3.4. Suposiciones y dependencias

Para una buena utilización y gestión de la aplicación WebHistoryFlow es aconsejable tener en cuenta las siguientes suposiciones y dependencias.

- La aplicación WebHistoryFlow, se ejecuta en el lenguaje de programación Java a través de su máquina virtual, con lo cual el servidor dónde se aloje la aplicación necesitará disponer de un Servidor Web como Apache y un Contenedor Web como Tomcat.
- El cliente podrá solicitar el suministro de recursos web a través de los distintos exploradores.

- Los equipos de lado del cliente deberán disponer de una JVM [24], Máquina Virtual de Java.
- Es obligatorio tener conexión a internet para la utilización de la aplicación WebHistoryFlow.
- Cuanto mayor sea el ancho de banda usado para la solicitud de recursos de la aplicación WebHistoryFlow, menor será el tiempo de recepción de los recursos.

### 3.5. Futuras versiones

Teniendo en cuenta que es la primera versión de WebHistoryflow podríamos decir que el numero de futuras versiones podría ser bastante amplio ya que en este proyecto se ha tratado los aspectos mas básicos. A continuación se relacionan algunas de ellas.

- Proporcionar resultados estadísticos relacionados con la interacción de datos entre los distintos autores o referidos a versiones concretas.
- Proporcionar datos para otros frameworks donde se visualicen distintos formatos gráficos.
- Mostrar los textos modificados de cada autor en cada versión .
- Establecer los resultados que ahora ofrece WebHistoryFlow en función de la versión que indique el cliente a través de otro parámetro. En esta versión de WebHistoryFlow se realiza en función de la última contribución.
- Implementar un plugin que realice el análisis gráfico de todos los ficheros pertenecientes a una svn o de un determinado grupo de ellos.
- Modularización de los recursos actualmente disponibles en bloques de aplicaciones, reduciendo el acoplamiento en la solicitud de recursos y aumentando los tiempos de respuesta.
- Gestión de los datos que generan algunos recursos desde buffers de memoria y no desde ficheros como se realiza en esta versión.
- Implementar un conjuntos de distintos gráficos estándar (sectores, barras etc...) para obtener otras formas de representación gráfica pero basándose en el tipo de información que tratar y procesar el plugin para wikis de History Flow.

## Capítulo 4

# Desarrollo general del proyecto

Para el buen desarrollo del proyecto, una correcta implementación y un seguimiento acorde a la metodología de desarrollo elegida, como se expone en el capítulo de introducción, se describe a continuación los apartados de análisis, diseño e implementación, dejando el apartado de evaluación y pruebas en el siguiente capítulo.

### 4.1. Análisis

Ahora se procederá al análisis para establecer el comportamiento deseado del sistema, los requisitos exigidos para describir que es lo que hace el sistema, y la definición de las estructuras de datos así como las relaciones existentes entre ellos.

#### 4.1.1. Identificación actores y casos de uso

En este proyecto por la forma en que se ha construido y teniendo en cuenta su función de suministrador de recursos, hay claramente definido un solo *Actor Principal*, que es el *Usuario Web o Cliente* que solicita recursos a la aplicación WebHistoryFlow.

Para ello definiremos los siguientes casos de uso:

- Visualizar Instrucciones de Uso
- Solicitar Gráfico History Flow Artículo Wiki
- Solicitar Gráfico History Flow Artículo Wiki de un Autor
- Solicitar Gráfico Gephi Artículo Wiki
- Solicitar Gráfico Gephi Artículo Wiki de un Autor
- Solicitar Datos Formato Gephi Artículo Wiki

- Solicitar Gráfico Prefuse Artículo Wiki
- Solicitar Gráfico Prefuse Artículo Wiki de un Autor
- Solicitar Datos Formato Prefuse Artículo Wiki
- Solicitar Gráfico History Flow Fichero Subversión
- Solicitar Gráfico History Flow Fichero Subversión de un Autor
- Solicitar Gráfico Gephi Fichero Subversión
- Solicitar Gráfico Gephi Fichero Subversión de un Autor
- Solicitar Datos Formato Gephi Fichero Subversión
- Solicitar Gráfico Prefuse Fichero Subversión
- Solicitar Gráfico Prefuse Fichero Subversión de un Autor
- Solicitar Datos Formato Prefuse Fichero Subversión

### 4.1.2. Modelado de casos de uso

A partir de estos diagramas se describirán las interacciones entre los actores externos y el sistema.

#### 4.1.2.1. Diagrama de casos de uso

Como realizaremos el análisis en función de un solo actor principal, a parte de mostrar el gráfico general *Figura 4.1* referido a los casos de uso del sistema, se añadirá a las correspondientes descripciones el gráfico particular con la relación *autor-caso de uso*.

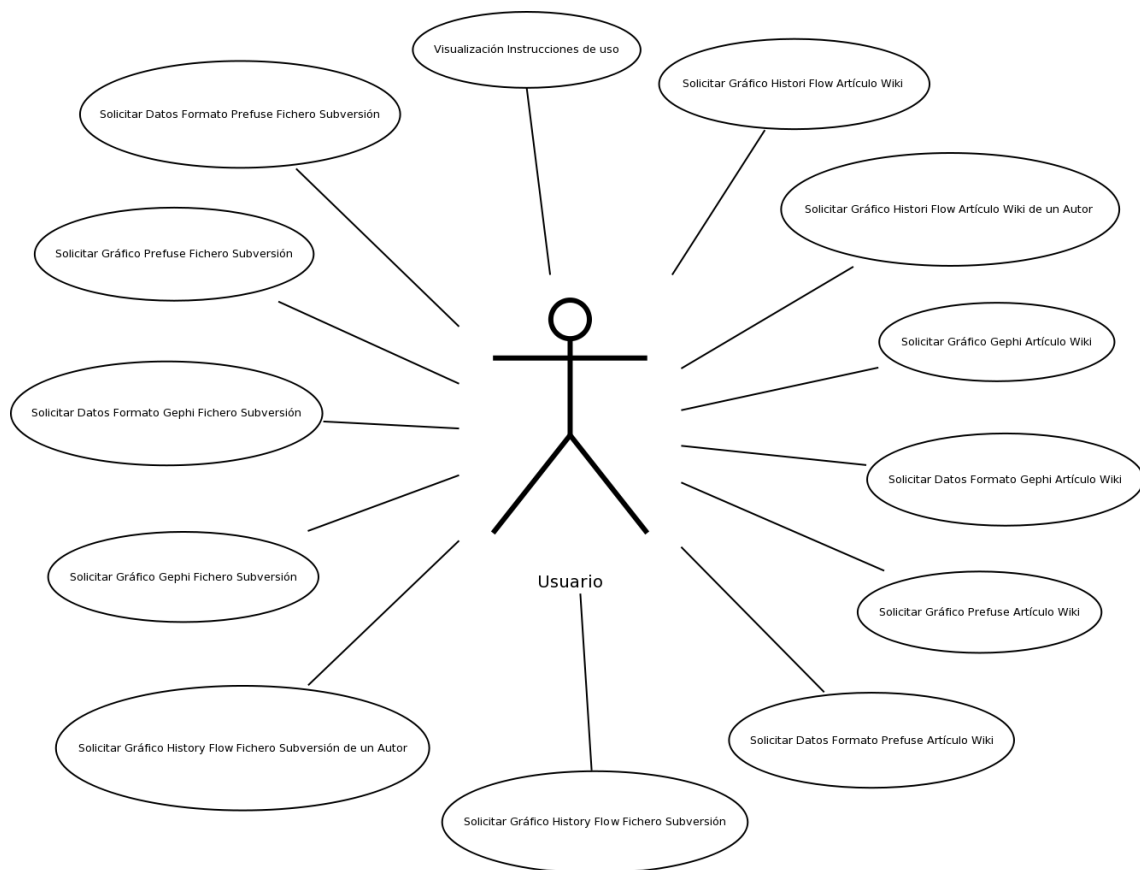
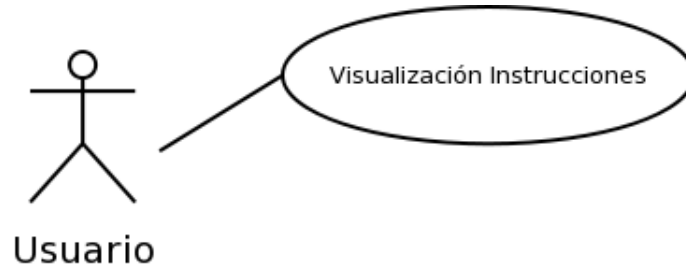


Figura 4.1: Casos de Uso pertenecientes al Sistema History Flow.

#### 4.1.2.2. Descripción casos de uso



**Caso de Uso :** Visualización de instrucciones.

**Descripción :** Visualización de las instrucciones de uso de la aplicación web.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita información de como se usa la aplicación.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección raíz de la URL del servidor.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* una página HTML con la información de uso de la aplicación.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** Los parámetros indicados no son correctos.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Gráfico History Flow Artículo Wiki.

**Descripción :** Devuelve gráfico con formato History Flow de un artículo wiki.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita gráfico History Flow de un artículo wiki.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor MediaWiki, tipo de gráfico, nombre de artículo y modo de visualización.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un gráfico History Flow con el flujo de información del artículo wiki.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor indicado para el artículo no es MediaWiki.

1. El *Sistema* indica el error.

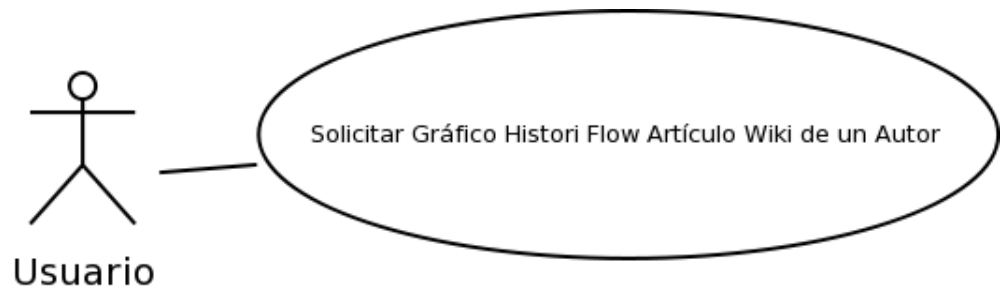
**3b.** El servidor MediaWiki indicado no contiene el artículo indicado.

1. El *Sistema* indica el error.

**3c.** El modo indicado no existe.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Gráfico History Flow Artículo Wiki de un Autor.

**Descripción :** Devuelve gráfico con formato History Flow de un artículo wiki perteneciente a un autor.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita gráfico History Flow de un artículo wiki referido a un autor concreto.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor MediaWiki, nombre de artículo, modo de visualización y autor.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un gráfico History Flow con el flujo de información del artículo wiki resaltando el flujo del autor indicado.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor indicado para el artículo no es MediaWiki.

1. El *Sistema* indica el error.

**3b.** El servidor MediaWiki indicado no contiene el artículo indicado.

1. El *Sistema* indica el error.

**3c.** El autor indicado no existe en la última versión.

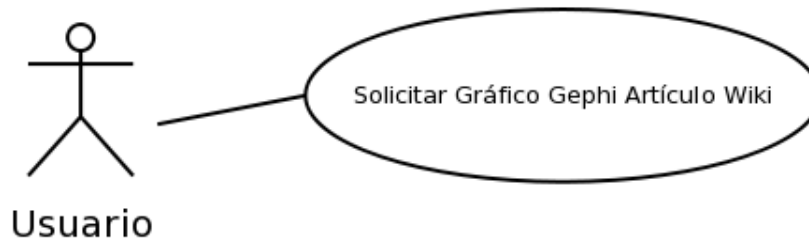
1. El *Sistema* indica el error.

**3d.** El modo indicado no existe.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.





**Caso de Uso :** Solicitar Gráfico Gephi Artículo Wiki.

**Descripción :** Devuelve gráfico con formato Gephi de un artículo wiki.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita gráfico Gephi de un artículo wiki.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor MediaWiki, tipo de gráfico y nombre de artículo.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un gráfico Gephi con el flujo de información del artículo wiki.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

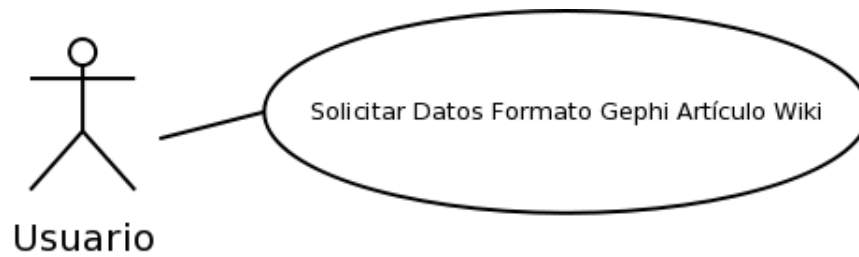
**3a.** El servidor indicado para el artículo no es MediaWiki.

1. El *Sistema* indica el error.

**3b.** El servidor MediaWiki indicado no contiene el artículo indicado.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Datos Formato Gephi Artículo Wiki.

**Descripción :** Devuelve el artículo wiki en formato *gml* de Gephi.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita una representación en formato *gml* de Gephi, del artículo wiki indicado.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, tipo de gráfico, servidor MediaWiki, nombre de artículo y formato de salida.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un fichero en formato *gml* de *Gephi*.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor indicado para el artículo no es MediaWiki.

1. El *Sistema* indica el error.

**3b.** El servidor MediaWiki indicado no contiene el artículo indicado.

1. El *Sistema* indica el error.

**3c.** El formato de salida no se corresponde con el tipo de gráfico indicado.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Gráfico Prefuse Artículo Wiki.

**Descripción :** Devuelve gráfico con formato Prefuse de un artículo wiki.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita gráfico Prefuse de un artículo wiki.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor MediaWiki, tipo de gráfico y nombre de artículo.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un gráfico Prefuse con la relación *autor-palabras* en forma de applet.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

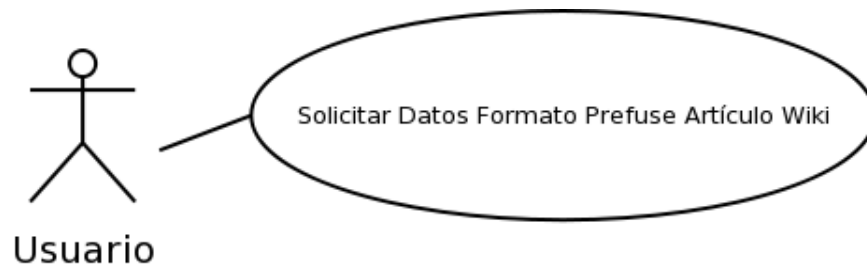
**3a.** El servidor indicado para el artículo no es MediaWiki.

1. El *Sistema* indica el error.

**3b.** El servidor MediaWiki indicado no contiene el artículo indicado.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Datos Formato Prefuse Artículo Wiki.

**Descripción :** Devuelve el artículo wiki en formato *xml* de Prefuse.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita una representación en formato *xml* de Prefuse, del artículo wiki indicado.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, tipo de gráfico, servidor MediaWiki, nombre de artículo y formato de salida.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un fichero en formato *xml de Prefuse*.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor indicado para el artículo no es MediaWiki.

1. El *Sistema* indica el error.

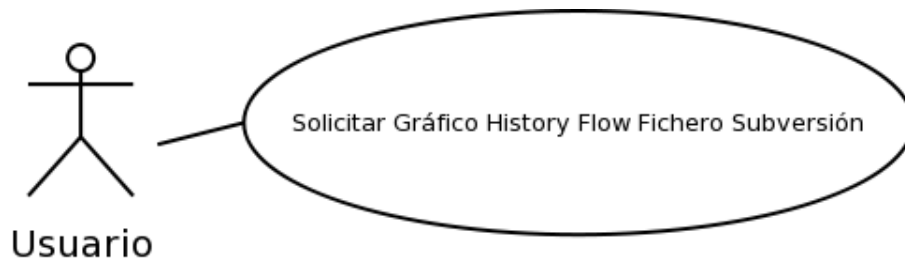
**3b.** El servidor MediaWiki indicado no contiene el artículo indicado.

1. El *Sistema* indica el error.

**3c.** El formato de salida no se corresponde con el tipo de gráfico indicado.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Gráfico History Flow Fichero Subversión.

**Descripción :** Devuelve gráfico con formato History Flow de un fichero perteneciente a una subversión.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita gráfico History Flow de un fichero que se encuentra en un servidor que almacena subversiones.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor svn, tipo de gráfico, nombre de fichero y modo de visualización.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un gráfico History Flow con el flujo de información del fichero perteneciente a la svn.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor indicado para el fichero no gestiona subversiones.

1. El *Sistema* indica el error.

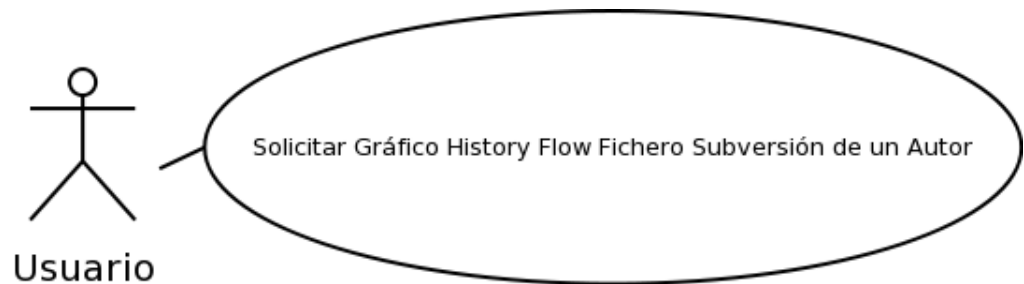
**3b.** El servidor svn indicado no contiene el fichero indicado.

1. El *Sistema* indica el error.

**3c.** El modo indicado no existe.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Gráfico History Flow Fichero Subversión de un Autor.

**Descripción :** Devuelve gráfico con formato History Flow de un fichero perteneciente a una subversión en función de uno de sus autores.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita gráfico History Flow de un fichero perteneciente a una svn, referido a un autor concreto.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor svn, nombre de fichero, modo de visualización y autor.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un gráfico History Flow con el flujo de información del fichero de la svn, resaltando el flujo del autor indicado.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor indicado para el fichero no gestiona subversiones.

1. El *Sistema* indica el error.

**3b.** El servidor svn indicado no contiene el fichero indicado.

1. El *Sistema* indica el error.

**3c.** El modo indicado no existe.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Gráfico Gephi Fichero Subversión.

**Descripción :** Devuelve gráfico con formato Gephi de un fichero perteneciente a una subversión.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita gráfico Gephi de un fichero que se encuentra en un servidor que almacena subversiones.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor svn, tipo de gráfico, nombre de fichero y modo de visualización.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un gráfico Gephi con el flujo de información del fichero perteneciente a la svn.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor svn indicado para el fichero no gestiona subversiones.

1. El *Sistema* indica el error.

**3b.** El servidor svn indicado no contiene el fichero indicado.

1. El *Sistema* indica el error.

**3c.** El modo indicado no existe.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Datos Formato Gephi Fichero Subversión.

**Descripción :** Devuelve un fichero de datos con formato *gml de Gephi* de un fichero perteneciente a una subversión.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita fichero de datos con formato *gml de Gephi* de un archivo perteneciente a una svn.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor svn, nombre de fichero, modo de visualización.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un fichero de datos con formato *gml de Gephi*.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor indicado para el fichero no gestiona subversiones.

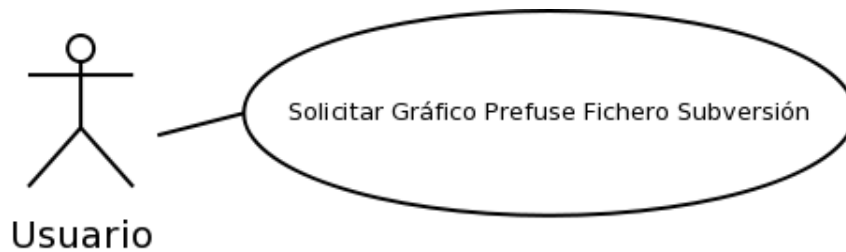
1. El *Sistema* indica el error.

**3b.** El servidor svn indicado no contiene el fichero indicado.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.





**Caso de Uso :** Solicitar Gráfico Prefuse Fichero Subversión.

**Descripción :** Devuelve gráfico con formato Prefuse de un fichero perteneciente a una subversión.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita gráfico Prefuse de un fichero que se encuentra en un servidor que almacena subversiones.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor svn, tipo de gráfico, nombre de fichero.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un gráfico Prefuse con la relación *autor-palabras* en forma de applet.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor svn indicado para el fichero no gestiona subversiones.

1. El *Sistema* indica el error.

**3b.** El servidor svn indicado no contiene el fichero indicado.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.



**Caso de Uso :** Solicitar Datos Formato Prefuse Fichero Subversión.

**Descripción :** Devuelve un fichero de datos con formato *xml de Prefuse* de un fichero perteneciente a una subversión.

**Actores :** Usuario Web.

**Resumen :** El usuario web solicita fichero de datos con formato *xml de Prefuse* de un archivo perteneciente a una svn.

**Escenario principal :**

1. El *Usuario Web* comienza escribiendo una dirección URL indicando los parámetros, servidor svn, nombre de fichero, modo de visualización.
2. El *Sistema* busca dentro de sus recursos para ver si esa ruta indicada está definida.
3. El *Sistema* encuentra dicha ruta y procesa los parámetros.
4. El *Sistema* devuelve al *Usuario Web* un fichero de datos con formato *xml de Prefuse*.

**Escenarios alternativos :**

**2a.** La ruta indicada no está definida.

1. El *Sistema* indica el error.

**3a.** El servidor indicado para el fichero no gestiona subversiones.

1. El *Sistema* indica el error.

**3b.** El servidor svn indicado no contiene el fichero indicado.

1. El *Sistema* indica el error.

**\*a** El *Usuario Web* en cualquier momento puede cancelar la visualización de información de uso de la aplicación.

### 4.1.3. Modelado conceptual de datos

#### 4.1.3.1. Diagrama de clases conceptuales

A través de este diagrama de clases se visualiza la estructura de la relación existente entre las distintas clases de la aplicación. Con el fin de mostrar un diagrama claro, se ha encapsulado algunas clases que representaban una mínima variación. Dividiré el diagrama en dos partes para mostrar con mas claridad los detalles del mismo y se comentará algunos detalles de las clases representadas.

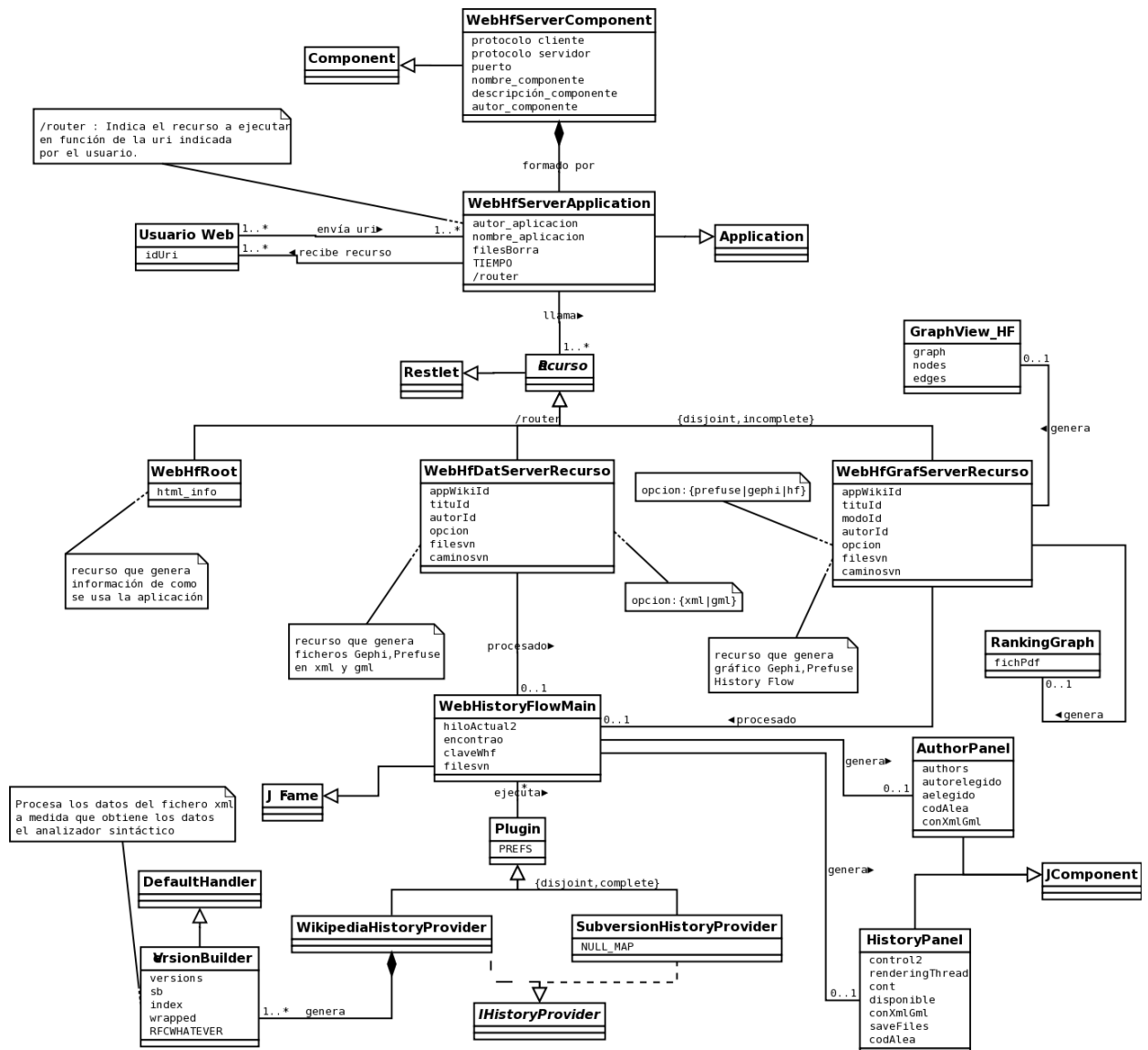


Figura 4.2: Diagrama de clases.

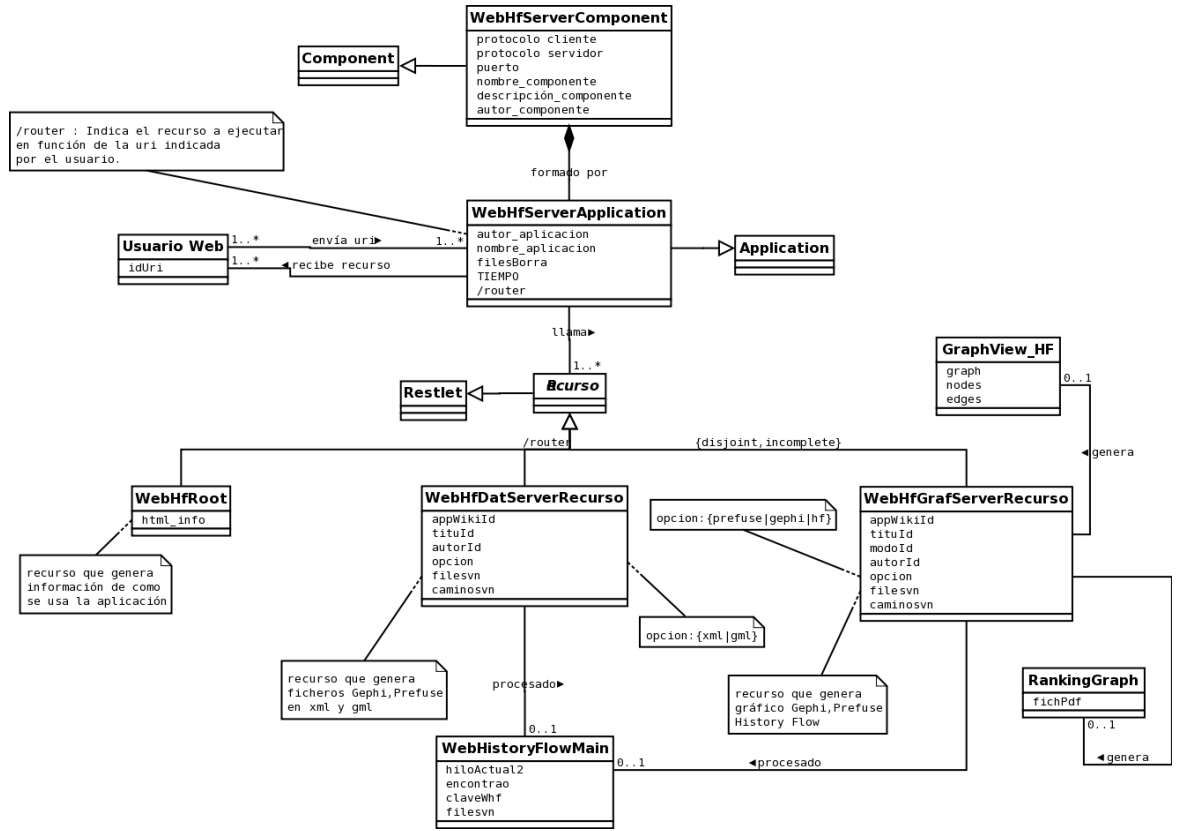


Figura 4.3: Diagrama de clases, bloque superior.

En la Figura 4.3 **WebHfServerApplication** es la clase encargada de llamar a los distintos recursos en función de la *URI* suministrada por el *Usuario Web*. La variable */router* es la que recibe la *URI* y en función de un mapeado previamente definido se accede al recurso correspondiente.

Las *clases recurso* **WebHfDatServerRecurso** y **WebHfGrafServerRecurso** encapsulan un conjunto de clases. A continuación se describen algunos detalles aclaratorios pertenecientes a los gráficos del Modelo Conceptual de Datos.

- La clase **WebHfRoot** de la Figura 4.3 es la que se encarga de generar como recurso una página *HTML* indicando la información de uso de la aplicación **WebHistoryFlow**.
- La clase **WebHfDatServerRecurso** de la Figura 4.3 encapsula aquellas clases que realizan la misma función excepto en el tipo de datos a visualizar. Los tipos que se encontraría agrupados en esta clase sería las salidas en formato *gml* y *xml*, tanto para *wikis* como para *subversiones*.
- La clase **WebHfGrafServerRecurso** de la Figura 4.3 agrupa aquellas clases que están encargadas de generar los distintos gráficos que la aplicación

WebhistoryFlow puede suministrar, que serían los pertenecientes a *Prefuse*, *Gephi* e *History Flow*, ya sean en función de una *wiki*, una *svn*, un autor concreto o un *modo de visualización* determinado.

- Las clases `GraphView_HF` y `RankingGraph` de la Figura 4.3 son las encargadas de generar, a partir de sus correspondientes API's, es decir *Prefuse* y *Gephi*, un gráfico tipo *applet* y un gráfico tipo *PDF*.
- Las clases `WikipediaHistoryProvider` y `SubversionHistoryProvider` de la Figura 4.4 son las encargadas de *extraer y procesar* los ficheros *xml* pertenecientes a un artículo *wiki* o a un fichero de una subversión.
- La clase `VersionBuilder` de la Figura 4.4 es la encargada de procesar las distintas *versiones* que posee el artículo *wiki* o el fichero de la subversión.
- Las clases `AuthorPanel` e `HistoryPanel` de la Figura 4.4 se encargan de generar todos los datos necesarios para suministrar el gráfico correspondiente de los autores y del flujo de información, ya sea de una *wiki* o de un fichero perteneciente a una *svn*.

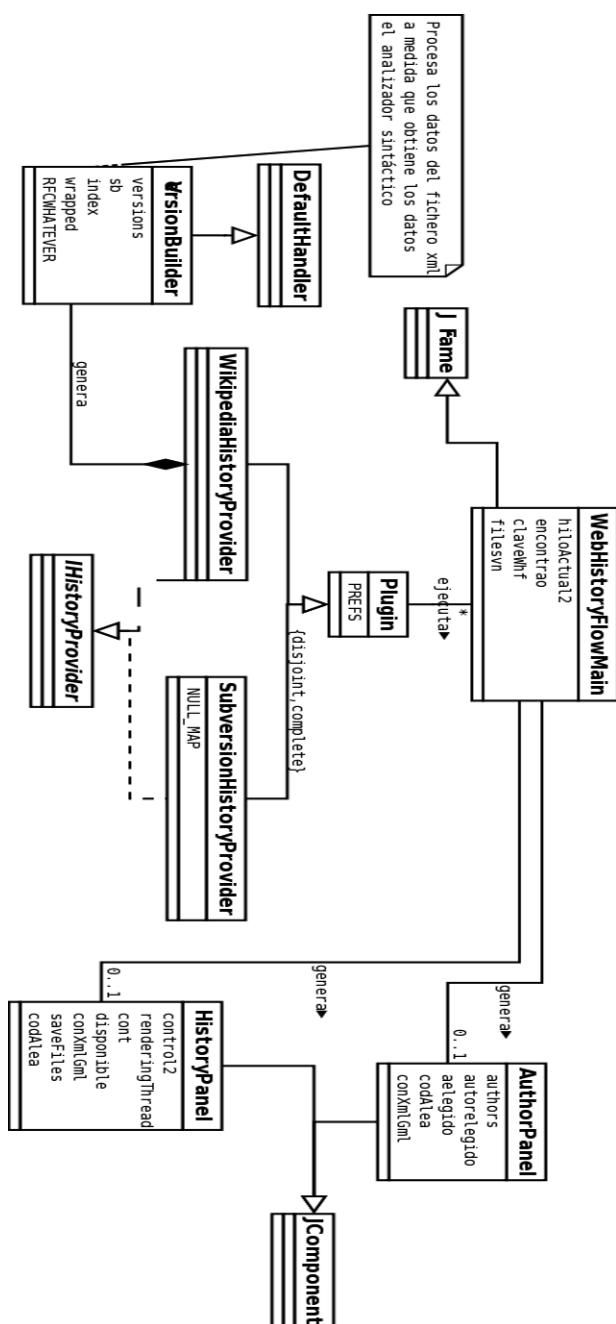


Figura 4.4: Diagrama de clases, bloque inferior.

#### 4.1.4. Modelo de comportamiento del sistema

##### 4.1.4.1. Diagramas de secuencia del sistema (DSS) y contrato de las operaciones

A través del DSS se representará la interacción como un gráfico bidimensional donde el eje vertical representará el tiempo avanzando descendentemente y horizontalmente representando los roles de colaboración entre los distintos objetos. A continuación de cada DSS se definirá su correspondiente contrato de las operaciones para describir el cambio de un estado a otro de los objetos del dominio como resultado de la ejecución de una operación.

##### DSS

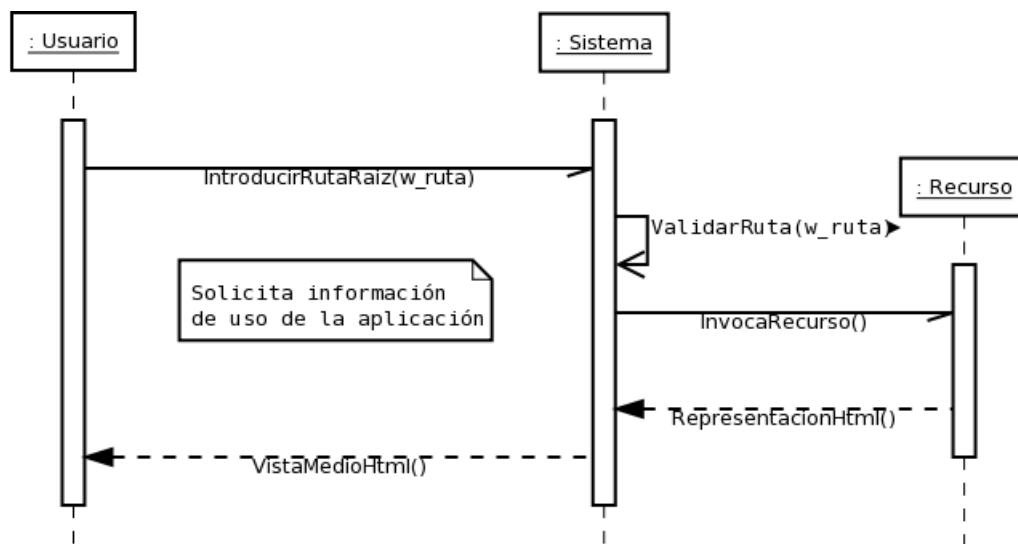


Figura 4.5: DSS, Caso uso: Visualización de instrucciones.

##### Contrato

**Operaci3n :** IntroducirRutaRaiz(w\_ruta)

**Responsabilidades :** Proporcionar informaci3n de uso de la aplicaci3n, donde router = w\_ruta

**Referencias Cruzadas :** Caso de Uso : Visualizaci3n de instrucciones

**Precondiciones :** Existe una aplicaci3n con router = w\_ruta

**Postcondiciones :**

- \* Se crea una instancia *aplicaci3n* de WebHfServerApplication.
- \* Se crea una instancia *recurso* de WebHfRoot.
- \* Se asign3 w\_ruta a aplicaci3n.router
- \* Se asoci3 el recurso solicitado *recurso* con *aplicaci3n*.

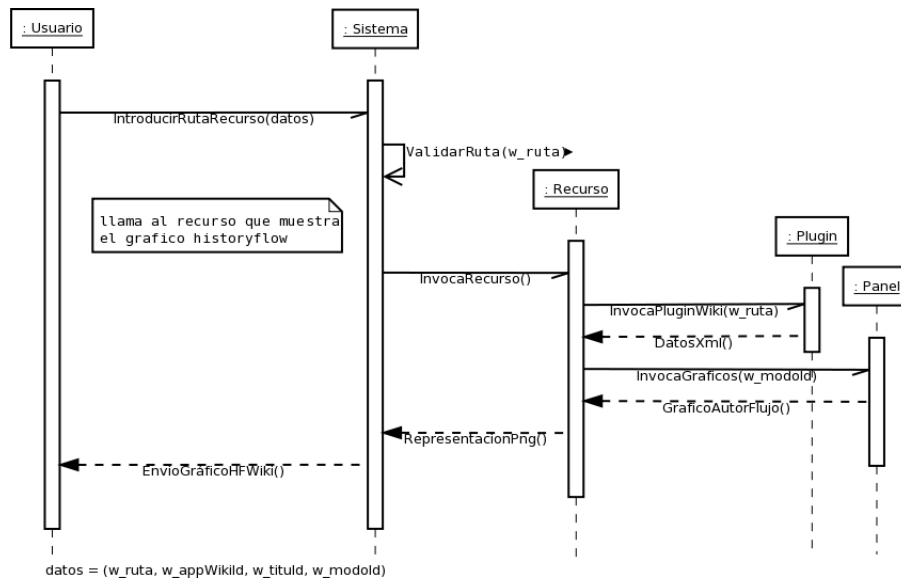
DSS

Figura 4.6: DSS, Caso uso: Solicitar Gráfico History Flow Artículo Wiki.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_appWikiId, w\_opcion, w\_tituloId, w\_modId)

**Responsabilidades :** Obtener recurso gráfico en formato History Flow donde router = w\_ruta

**Referencias Cruzadas :** Caso de Uso : Solicitar Gráfico History Flow Artículo Wiki

**Precondiciones :** Existe una aplicación con router = w\_ruta, opción = "hf"

**Postcondiciones :**

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recurso* de WebHfGrafServerRecurso.
- \* Se crea una instancia *pluginwiki* de WikipediaHistoryProvider.
- \* Se crea una instancia *panelAutor* de HistoryPanel, *panelFlujo* de AuthorPanel.
- \* Se asoció el plugin *pluginwiki* con *aplicación*.
- \* Se asignó w\_appWikiId a recurso.appWikiId, w\_tituloId a recurso.tituloId, w\_modId a recurso.modId
- \* Se asoció el recurso solicitado *recurso* con *aplicación*.
- \* Se asoció el recurso solicitado *panelAutor* y *panelFlujo* con *recurso*.



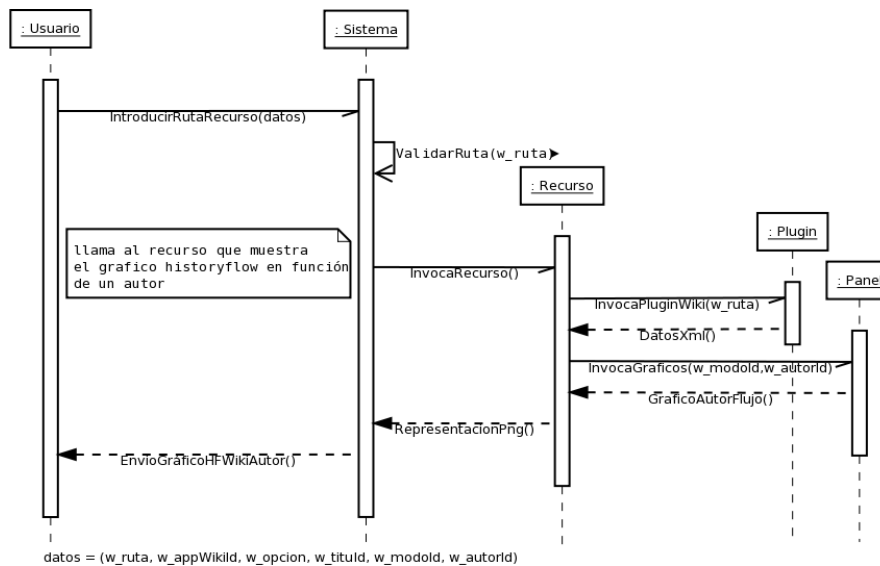
DSS

Figura 4.7: DSS, Caso uso: Solicitar Gráfico History Flow Artículo Wiki de un Autor.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_appWikiId, w\_opcion , w\_tituld, w\_modold, w\_autorId)

**Responsabilidades :** Obtener recurso que muestra gráfico History Flow en función de un autor y a partir de un artículo wiki, donde router = w\_ruta, autorId = w\_autorId.

**Referencias Cruzadas :** Caso de Uso : Solicitar Gráfico History Flow Artículo Wiki de un Autor

**Precondiciones :** Existe una aplicación con ruta = w\_ruta, un recurso con autorId = w\_autor , opción = "hf"

**Postcondiciones :**

- \* Se crea una instancia aplicación de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router.
- \* Se crea una instancia recurso de WebHfGrafServerRecurso.
- \* Se crea una instancia *pluginwiki* de WikipediaHistoryProvider.
- \* Se asoció el plugin *pluginwiki* con *aplicación*.
- \* Se crea una instancia *panelAutor* de HistoryPanel, *panelFlujo* de AuthorPanel.
- \* Se asoció el recurso solicitado *panelAutor* y *panelFlujo* con *recurso*.
- \* Se asignó w\_modold a recurso.modold, w\_autorId a recurso.autorId, w\_tituld a recurso.tituld
- \* Se asoció el recurso solicitado *recurso* con *aplicación*.

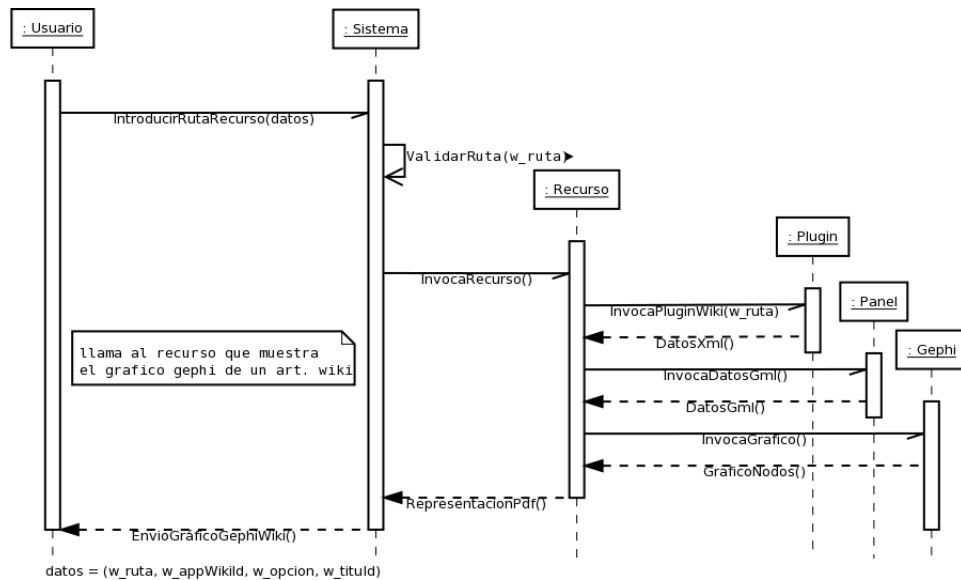
DSS

Figura 4.8: DSS, Caso uso: Solicitar Gráfico Gephi Artículo Wiki.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_appWikiId, w\_opcion , w\_tituloId)

**Responsabilidades :** Obtener recurso que muestra gráfico en formato Gephi de un artículo perteneciente a una wiki, donde router = w\_ruta

**Referencias Cruzadas :** Caso de Uso : Solicitar Gráfico Gephi Artículo Wiki

**Precondiciones :** Existe un recurso con router = w\_ruta, opción = "gephi"

**Postcondiciones :**

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recurso* de WebHfGrafServerRecurso.
- \* Se asignó w\_appWikiId a recurso.appWikiId, w\_tituloId a recurso.tituloId
- \* Se asoció el recurso solicitado *recurso* con *aplicación*.
- \* Se crea una instancia *pluginwiki* de WikipediaHistoryProvider.
- \* Se crea una instancia *panelAutor* de HistoryPanel, *panelFlujo* de AuthorPanel.
- \* Se asoció el plugin *pluginwiki* con *aplicación*.
- \* Se crea una instancia *grafgephi* de RankingGraph.
- \* Se asoció el gráfico *grafgephi* con *recurso*.
- \* Se asoció el recurso solicitado *panelAutor* y *panelFlujo* con *recurso*.

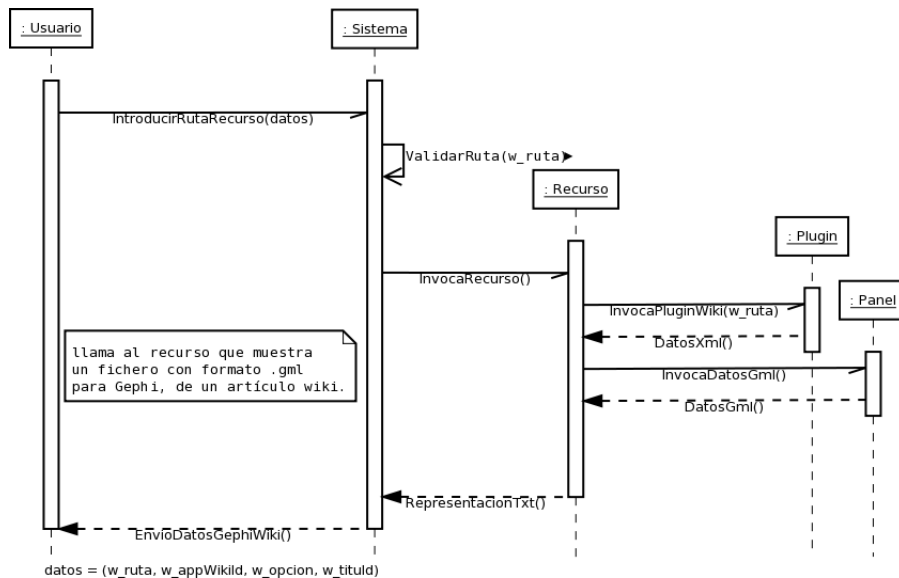
DSS

Figura 4.9: DSS, Caso uso: Solicitar Datos Formato Gephi Artículo Wiki.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_appWikiId, w\_titulo, w\_opcion)

**Responsabilidades :** Obtener recurso que genera fichero datos en formato gephi de un artículo wiki, donde router = w\_ruta

**Referencias Cruzadas :** Caso de Uso : Solicitar Datos Formato Gephi Artículo Wiki

**Precondiciones :** Existe una aplicación con router = w\_ruta y un recurso con opción = "gml".

**Postcondiciones :**

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recursodat* de WebHfDatServerRecurso.
- \* Se asignó w\_appWikiId a recursodat.appWikiId, w\_titulo a recursodat.titulo, w\_opcion a recursodat.opcion
- \* Se asoció el recurso solicitado *recursodat* con *aplicación*.
- \* Se crea una instancia *pluginwiki* de WikipediaHistoryProvider.
- \* Se asoció el plugin *pluginwiki* con *aplicación*.
- \* Se crea una instancia *datosgml* de HistoryPanel.
- \* Se asoció los datos Gephi *datosgml* con *recursodat*.

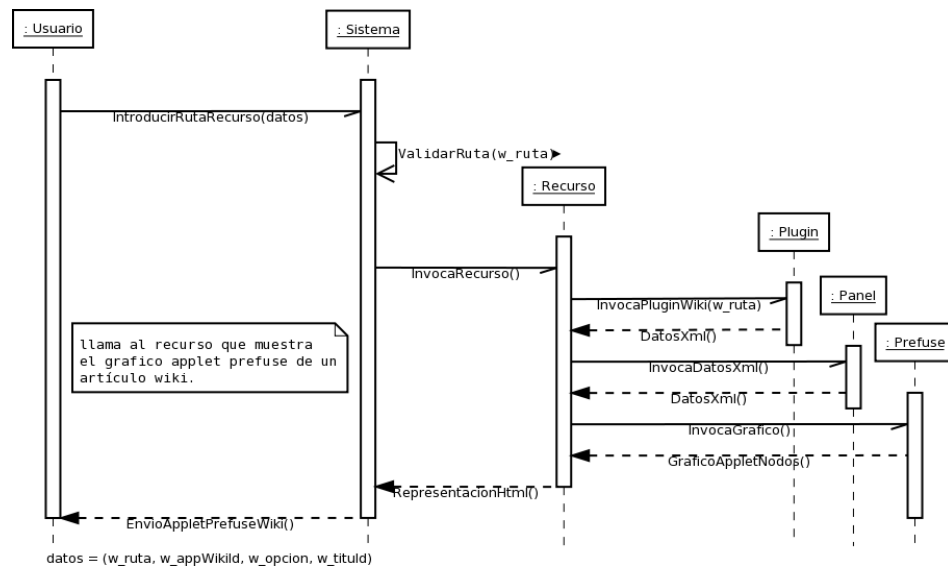
DSS

Figura 4.10: DSS, Caso uso: Solicitar Gráfico Prefuse Artículo Wiki.

Contrato

**Operación** : IntroducirRutaRecurso(w\_ruta, w\_opcion, w\_appWikiId, w\_tituloId)

**Responsabilidades** : Obtener recurso que genera gráfico Prefuse en forma de applet en función de un artículo wiki, donde ruta = w\_ruta

**Referencias Cruzadas** : Caso de Uso : Solicitar Gráfico Prefuse Artículo Wiki

**Precondiciones** : Existe un recurso con ruta = w\_ruta, recurso con opcion = "prefuse"

**Postcondiciones** :

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recurso* de WebHfGrafServerRecurso.
- \* Se asignó w\_appWikiId a recurso.appWikiId , w\_tituloId a recurso.tituloId
- \* Se crea una instancia *panelAutor* de HistoryPanel, *panelFlujo* de AuthorPanel.
- \* Se asoció el recurso solicitado *panelAutor* y *panelFlujo* con *recurso*.
- \* Se asoció el recurso solicitado *recurso* con *aplicación*.
- \* Se crea una instancia *pluginwiki* de WikipediaHistoryProvider.
- \* Se asoció el plugin *pluginwiki* con *aplicación*.
- \* Se crea una instancia *grafprefuse* de GraphView\_HF.
- \* Se asoció el gráfico Prefuse *grafprefuse* con *recurso*.

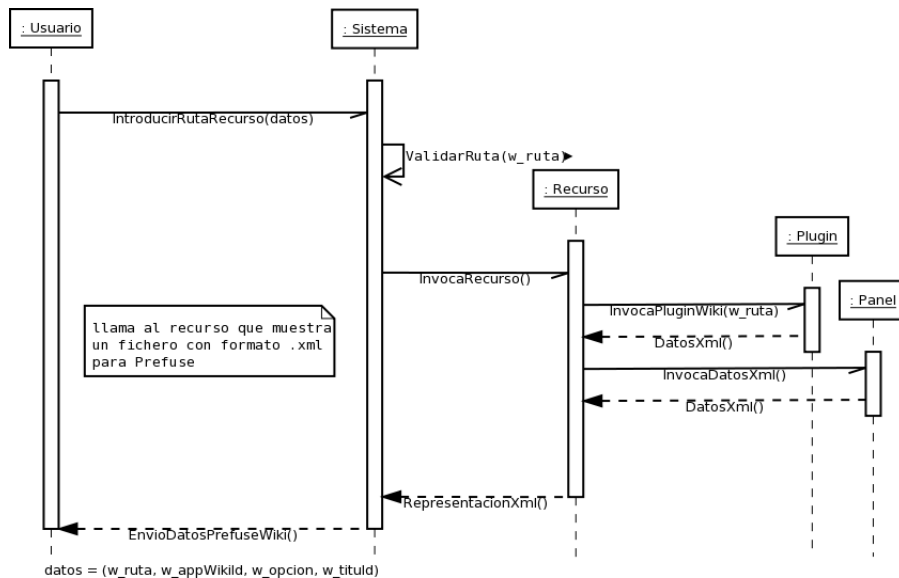
DSS

Figura 4.11: DSS, Caso uso: Solicitar Datos Formato Prefuse Artículo Wiki.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_opcion, w\_appWikiId, w\_titulo)

**Responsabilidades :** Obtener recurso que genera fichero datos en formato prefuse en función de un artículo wiki, donde ruta=w\_ruta

**Referencias Cruzadas :** Caso de Uso : Solicitar Datos Formato Prefuse Artículo Wiki

**Precondiciones :** Existe una aplicación con router = w\_ruta, un recursodat con opcion = "xml"

**Postcondiciones :**

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recursodat* de WebHfDatServerRecurso.
- \* Se asignó w\_appWikiId a recursodat.appWikiId, w\_titulo a recursodat.titulo
- \* Se asoció el recurso solicitado *recursodat* con *aplicación*.
- \* Se crea una instancia *pluginwiki* de WikipediaHistoryProvider.
- \* Se asoció el plugin *pluginwiki* con *aplicación*.
- \* Se crea una instancia *datosxml* de HistoryPanel.
- \* Se asoció los datos Prefuse *datosxml* con *recursodat*.

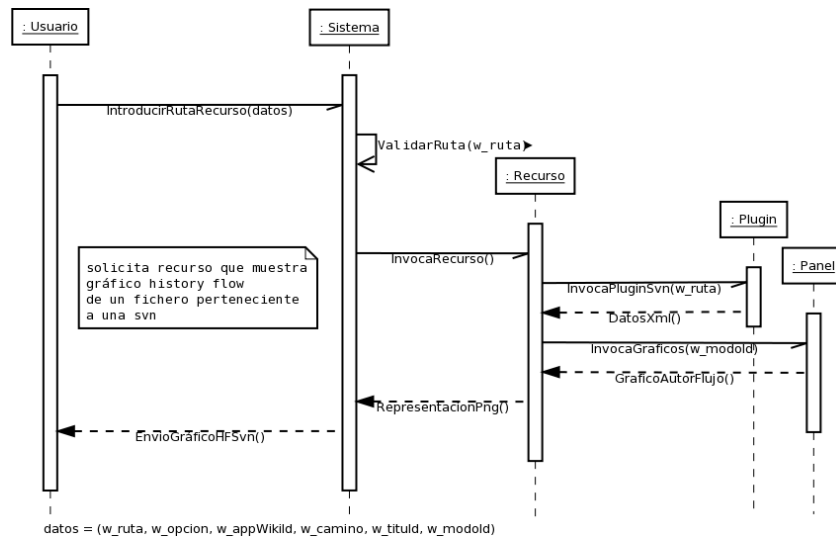
DSS

Figura 4.12: DSS, Caso uso: Solicitar Gráfico History Flow Fichero Subversión.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_opcion, w\_appWikiId, w\_caminoId, w\_tituloId, w\_modoid)

**Responsabilidades :** Obtener recurso con formato gráfico History Flow de un fichero de una svn, donde router = w\_ruta

**Referencias Cruzadas : Caso de Uso :** Solicitar Gráfico History Flow Fichero Subversión

**Precondiciones :** Existe una aplicación con router = w\_ruta, un recurso con modo = w\_mod, opcion = "hf"

**Postcondiciones :**

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recurso* de WebHfGrafServerRecurso.
- \* Se asignó w\_appWikiId a recurso.appWikiId, w\_caminoId a recurso.caminoId, w\_tituloId a recurso.tituloId, w\_modoid a recurso.modoid
- \* Se crea una instancia *recurso* de WebHfGrafServerRecurso.
- \* Se asignó w\_modoid a recurso.modoid, w\_autorId a recurso.autorId, w\_tituloId a recurso.tituloId
- \* Se asoció el recurso solicitado *recurso* con *aplicación*.
- \* Se crea instancia *panelAutor* de HistoryPanel, *panelFlujo* de AuthorPanel.
- \* Se asoció el recurso solicitado *panelAutor* y *panelFlujo* con *recurso*.
- \* Se crea una instancia *pluginsvn* de SubversionHistoryProvider.
- \* Se asoció el plugin *pluginsvn* con *aplicación*.

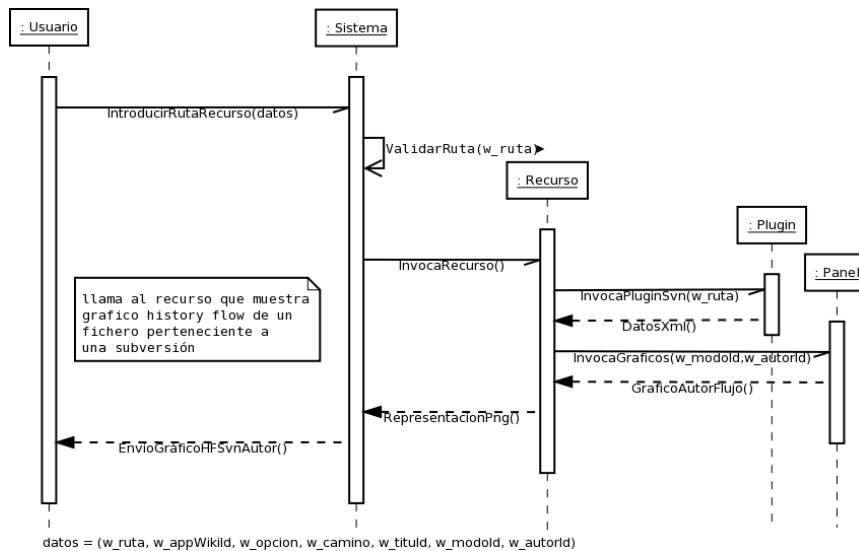
DSS

Figura 4.13: DSS, Caso uso: Solicitar Gráfico History Flow Fichero Subversión de un Autor.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_opcion, w\_appWikiId, w\_caminoId, w\_tituloId, w\_modold, w\_autorId)

**Responsabilidades :** Obtener recurso en formato gráfico History Flow de un autor perteneciente a un fichero de una svn, donde router = w\_ruta, autorId = w\_autorId

**Referencias Cruzadas : Caso de Uso :** Solicitar Gráfico History Flow Fichero Subversión de un Autor

**Precondiciones :** Existe una aplicación con router = w\_ruta, un recurso con modo = w\_modold, opcion = "hf"

**Postcondiciones :**

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recurso* de WebHfGrafServerRecurso.
- \* Se asignó w\_appWikiId a recurso.appWikiId, w\_caminoId a recurso.caminoId, w\_tituloId a recurso.tituloId, w\_modold a recurso.modold, w\_autorId a recurso.autorId
- \* Se asoció el recurso solicitado *recurso* con *aplicación*.
- \* Se crea una instancia *pluginsvn* de SubversionHistoryProvider.
- \* Se asoció el plugin *pluginsvn* con *aplicación*.
- \* Se crea instancia *panelAutor* de HistoryPanel, *panelFlujo* de AuthorPanel.
- \* Se asoció el recurso solicitado *panelAutor* y *panelFlujo* con *recurso*.

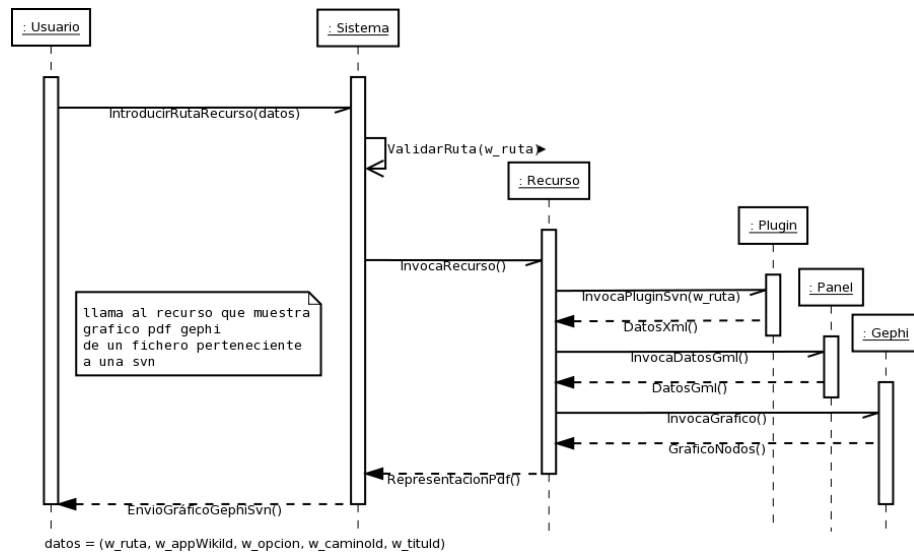
DSS

Figura 4.14: DSS, Caso uso: Solicitar Gráfico Gephi Fichero Subversión.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_opcion, w\_appWikiId, w\_caminoid, w\_titulid)

**Responsabilidades :** Obtener recurso con formato gráfico gephi para un fichero de una svn, donde router = w\_ruta

**Referencias Cruzadas :** Caso de Uso : Solicitar Gráfico Gephi Fichero Subversión

**Precondiciones :** Existe una aplicación con router = w\_ruta, un recurso con opcion = "gephi"

**Postcondiciones :**

- \* Se crea una instancia aplicación de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia recurso de WebHfGrafServerRecurso.
- \* Se asignó w\_appWikiId a recurso.appWikiId , w\_caminoid a recurso.caminoid, w\_titulid a recurso.titulid
- \* Se asoció el recurso solicitado *recurso* con *aplicación*.
- \* Se crea instancia *panelAutor* de HistoryPanel, *panelFlujo* de AuthorPanel.
- \* Se asoció el recurso solicitado *panelAutor* y *panelFlujo* con *recurso*.
- \* Se crea una instancia *pluginsvn* de SubversionHistoryProvider.
- \* Se asoció el plugin *pluginsvn* con *aplicación*.
- \* Se crea una instancia *grafgephi* de RankingGraph.
- \* Se asoció el gráfico *grafgephi* con *recurso*.



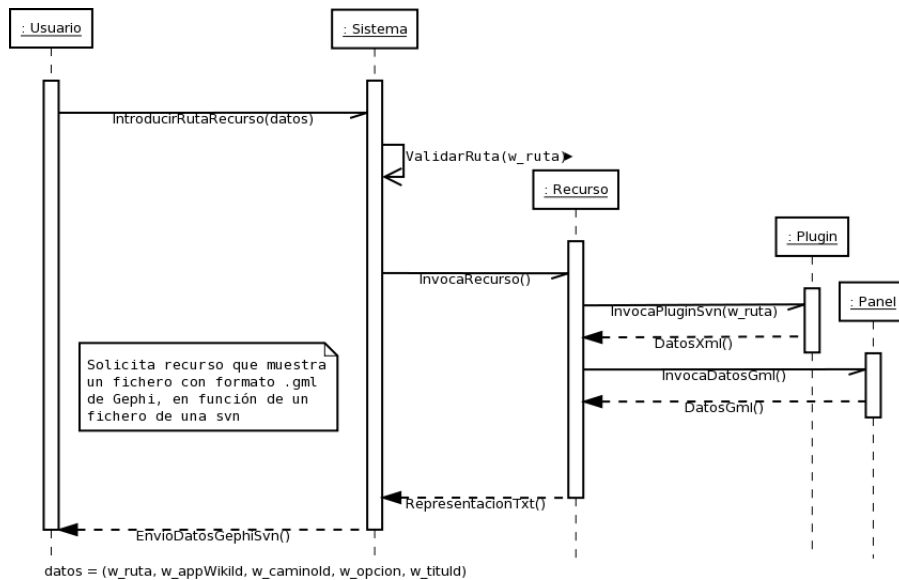
DSS

Figura 4.15: DSS, Caso uso: Solicitar Datos Formato Gephi Fichero Subversión.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_appWikiId, w\_caminoid, w\_titulId, w\_opcion)

**Responsabilidades :** Obtener recurso fichero de datos en formato gephi de un fichero perteneciente a una svn, donde router = w\_ruta

**Referencias Cruzadas :** **Caso de Uso :** Solicitar Datos Formato Gephi Fichero Subversión

**Precondiciones :** Existe una aplicación con router = w\_ruta, un recursodat con opción = "gml"

**Postcondiciones :**

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recursodat* de WebHfDatServerRecurso.
- \* Se asignó w\_appWikiId a recursodat.appWikiId, w\_titulId a recursodat.titulId, w\_caminoid a recursodat.caminoid
- \* Se crea una instancia *pluginsvn* de SubversionHistoryProvider.
- \* Se asoció el plugin *pluginsvn* con *aplicación*.
- \* Se crea una instancia *datosgml* de HistoryPanel.
- \* Se asoció los datos Gephi *datosgml* con *recursodat*.

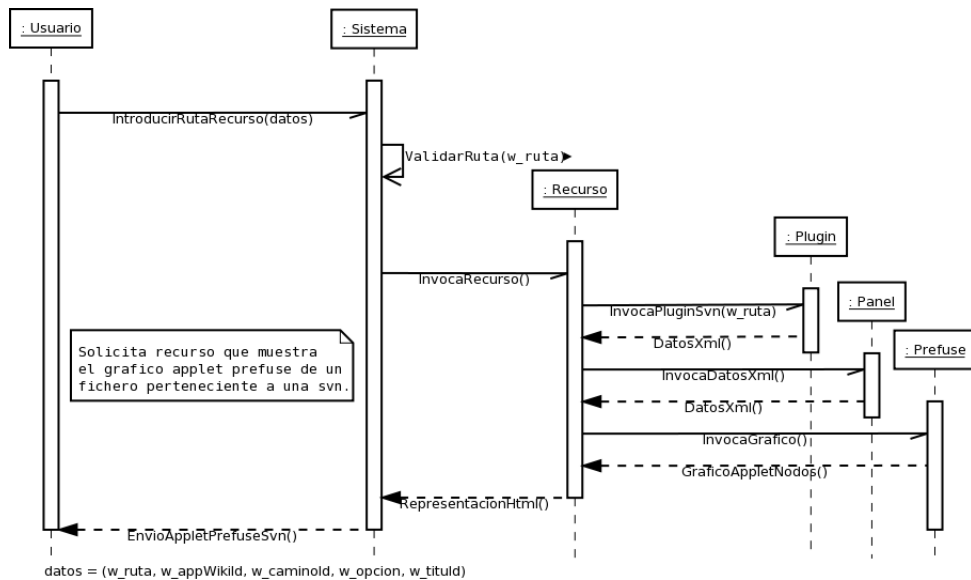
DSS

Figura 4.16: DSS, Caso uso: Solicitar Gráfico Prefuse Fichero Subversión.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_appWikiId, w\_caminoid, w\_tituld, w\_opcion)

**Responsabilidades :** Obtener recurso con formato gráfico prefuse para un fichero de una svn, donde router = w\_ruta

**Referencias Cruzadas :** Caso de Uso : Solicitar Gráfico Prefuse Fichero Subversión

**Precondiciones :** Existe una aplicación con router = w\_ruta, un recurso con opción = "prefuse"

**Postcondiciones :**

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recurso* de WebHfGrafServerRecurso.
- \* Se asignó w\_appWikiId a recurso.appWikiId , w\_caminoid a recurso.caminoid, w\_tituld a recurso.tituld
- \* Se crea instancia *panelAutor* de HistoryPanel, *panelFlujo* de AuthorPanel.
- \* Se asoció el recurso solicitado *panelAutor* y *panelFlujo* con *recurso*.
- \* Se crea una instancia *pluginsvn* de SubversionHistoryProvider.
- \* Se asoció el plugin *pluginsvn* con *aplicación*.
- \* Se crea una instancia *grafprefuse* de GraphView\_HF.
- \* Se asoció el gráfico Prefuse *grafprefuse* con *recurso*.

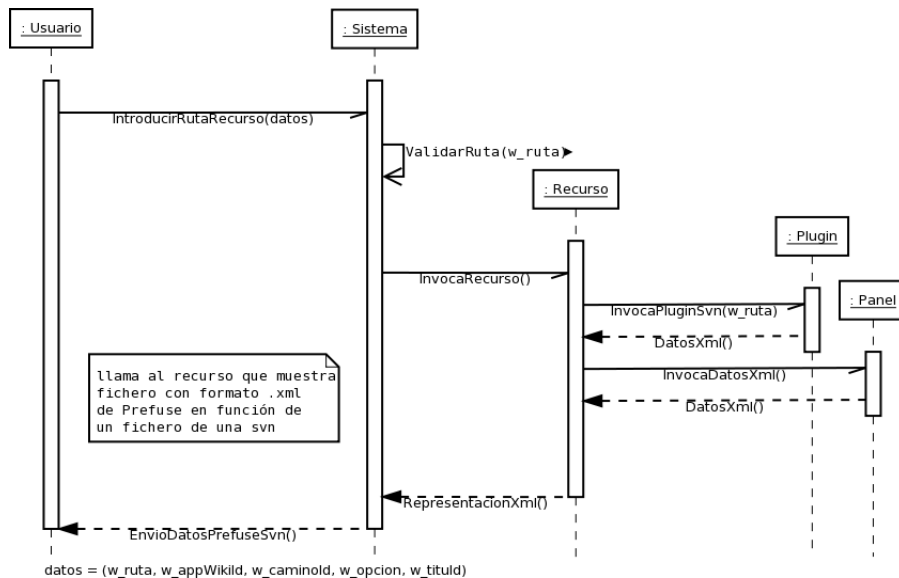
DSS

Figura 4.17: DSS, Caso uso: Solicitar Datos Formato Prefuse Fichero Subversión.

Contrato

**Operación :** IntroducirRutaRecurso(w\_ruta, w\_appWikiId, w\_caminoid, w\_titulid, w\_opcion)

**Responsabilidades :** Obtener recurso fichero datos con formato prefuse de un fichero de una svn, donde router = w\_ruta

**Referencias Cruzadas :** Caso de Uso : Solicitar Datos Formato Prefuse Fichero Subversión

**Precondiciones :** Existe una aplicación con router = w\_ruta, un recursodat con opción = "xml"

**Postcondiciones :**

- \* Se crea una instancia *aplicación* de WebHfServerApplication.
- \* Se asignó w\_ruta a aplicación.router
- \* Se crea una instancia *recursodat* de WebHfDatServerRecurso.
- \* Se asignó w\_appWikiId a recursodat.appWikiId, w\_titulid a recursodat.titulid, w\_caminoid a recursodat.caminoid
- \* Se asoció el recurso solicitado *recursodat* con *aplicación*.
- \* Se crea una instancia *pluginsvn* de WikipediaHistoryProvider.
- \* Se asoció el plugin *pluginsvn* con *aplicación*.
- \* Se crea una instancia *datosxml* de HistoryPanel.
- \* Se asoció los datos Prefuse *datosxml* con *recursodat*.

## 4.2. Diseño

En el apartado de diseño no se va a decir qué hace el sistema, sino como lo hace, describiéndose como se va a estructurar para que haga lo que tiene que hacer. Se representa para ello los diagramas de interacción necesarios así como el diseño de las clases y las asociaciones entre ellas.

### 4.2.1. Diagramas de interacción o colaboración

Para establecer las asociaciones entre objetos y los mensajes que se pasan de un objeto a otro, se representa a continuación una serie de *diagramas de interacción* o *diagramas de colaboración* en función de los distintos *casos de uso* definidos en apartados anteriores [4.1].

#### 4.2.1.1. Diagrama de colaboración CU Visualización de Instrucciones de Uso

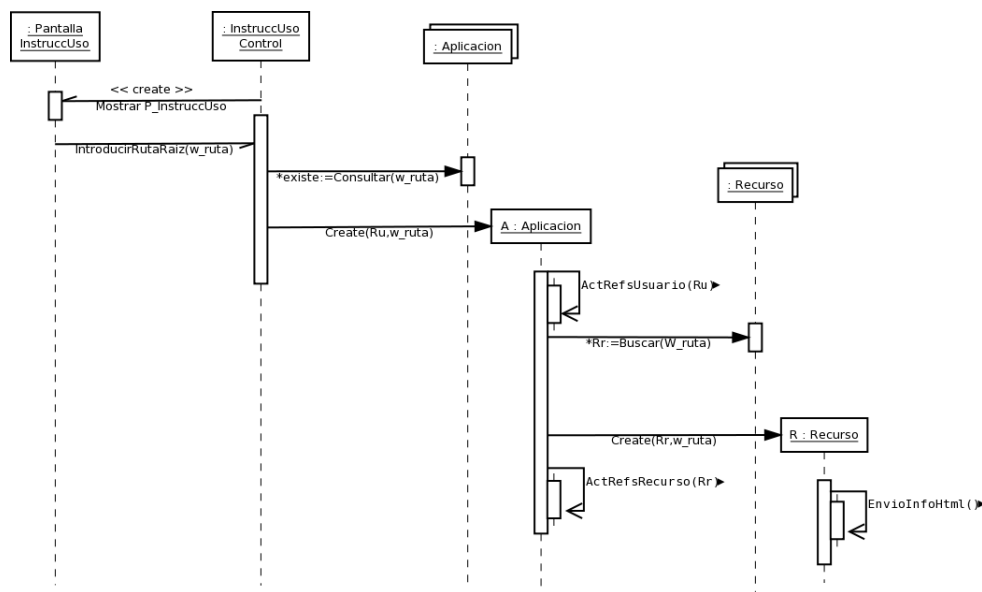


Figura 4.18: Diagrama de interacción, C.U.: Visualización Instrucciones de Uso

Observaciones a tener en cuenta en el diagrama de la Figura 4.18 :

- El motivo del uso de conjunto de objetos tanto en *Aplicación* como en *Recurso* es para representar por ejemplo en el caso de *Aplicación*, la posibilidad de creación de varias aplicaciones para un mismo cliente y a su vez cada cliente puede ejecutar distintos recursos, todos ellos en modo concurrente si así lo necesitara, ya que como se comentó anteriormente las clases desde las que heredan poseen esta característica.

- Usaremos *Aplicación* para designar la clase o un objeto de la clase *WebHf-ServerApplication*.
- *\*existe:=Consultar()*, comprobará la existencia del recurso en el servidor a través de la ruta indicada desde la URI.
- Los mensajes *ActRefsUsuario()* y *ActRefsRecurso()*, indican las asociaciones entre *Aplicación-Usuario* y *Aplicación-Recurso* respectivamente.
- Finalmente el objeto *R* de *Recurso* se encarga de generar la página en formato HTML con la información de uso que es la que será devuelta al cliente desde la petición que realizó.

#### 4.2.1.2. Diagrama de colaboración de CU con formato de salida History Flow artículo wiki

Debido a la longitud del diagrama de interacción de este caso de uso a parte de visualizar una vista general de dicho diagrama, descrito en la Figura 4.22, se hará una división en tres partes para su explicación y para tener una vista ampliada de cada una de las partes del diagrama.

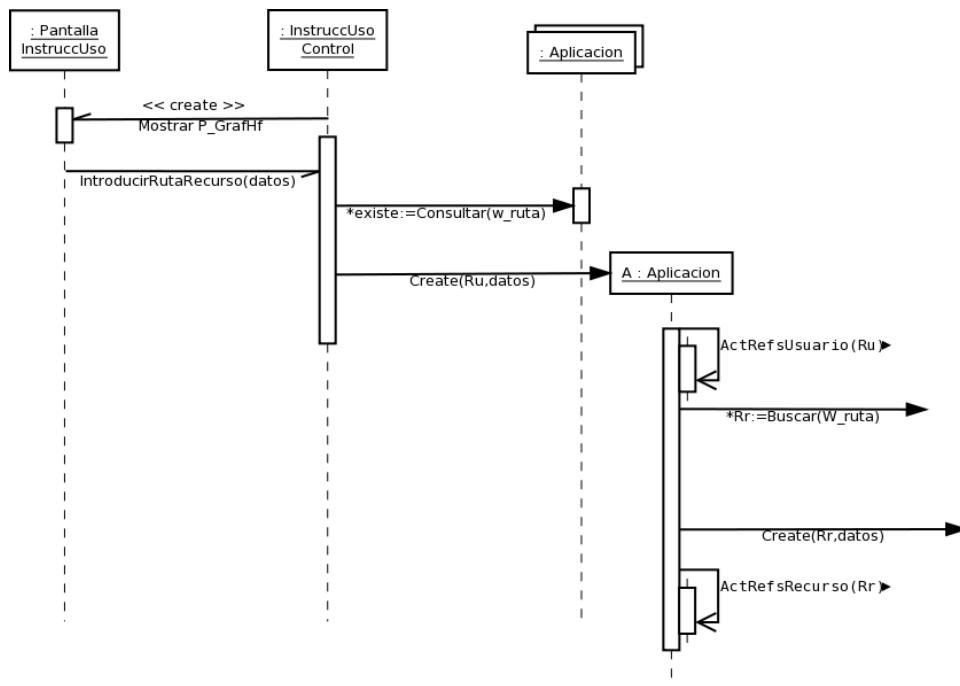


Figura 4.19: Diagrama de interacción, C.U.: Solicitar Gráfico History Flow Artículo Wiki, bloque superior figura 4.22

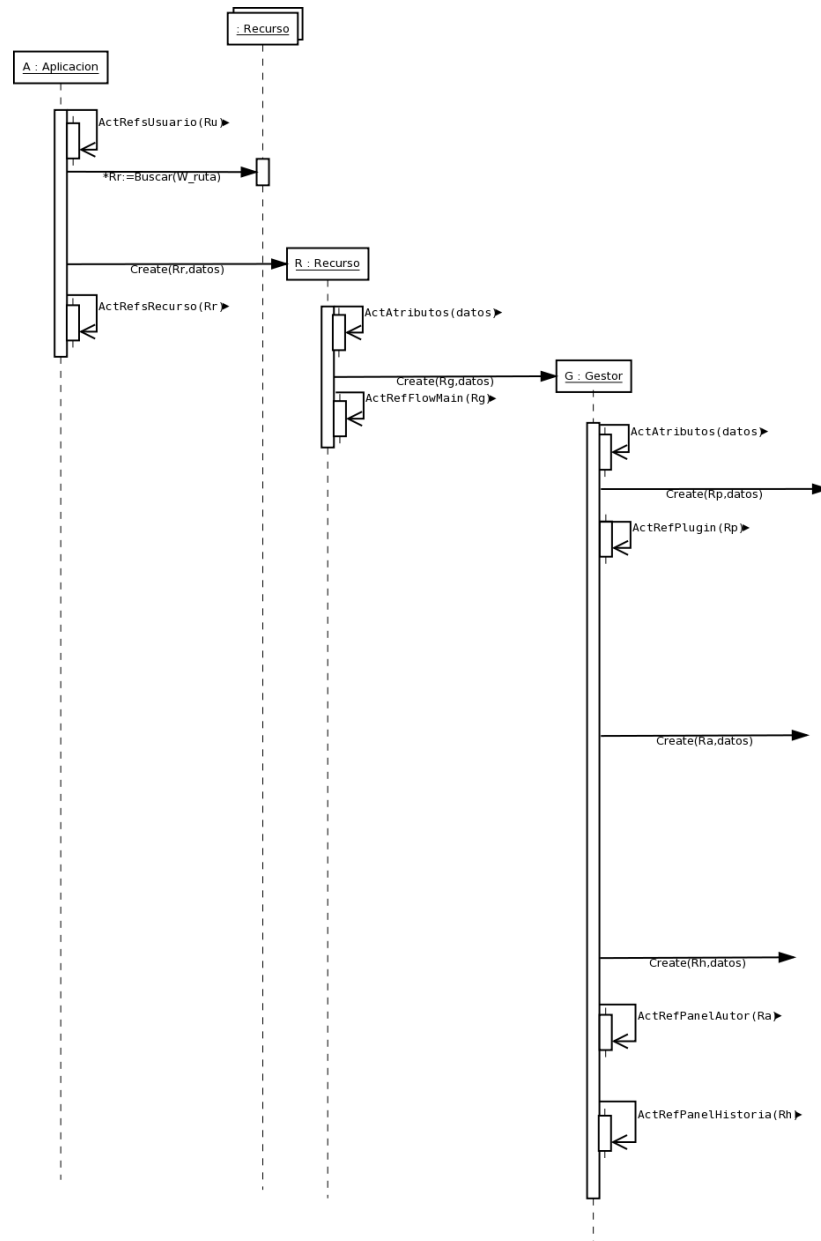


Figura 4.20: Diagrama de interacción, C.U.: Solicitar Gráfico History Flow Artículo Wiki, bloque medio figura 4.22

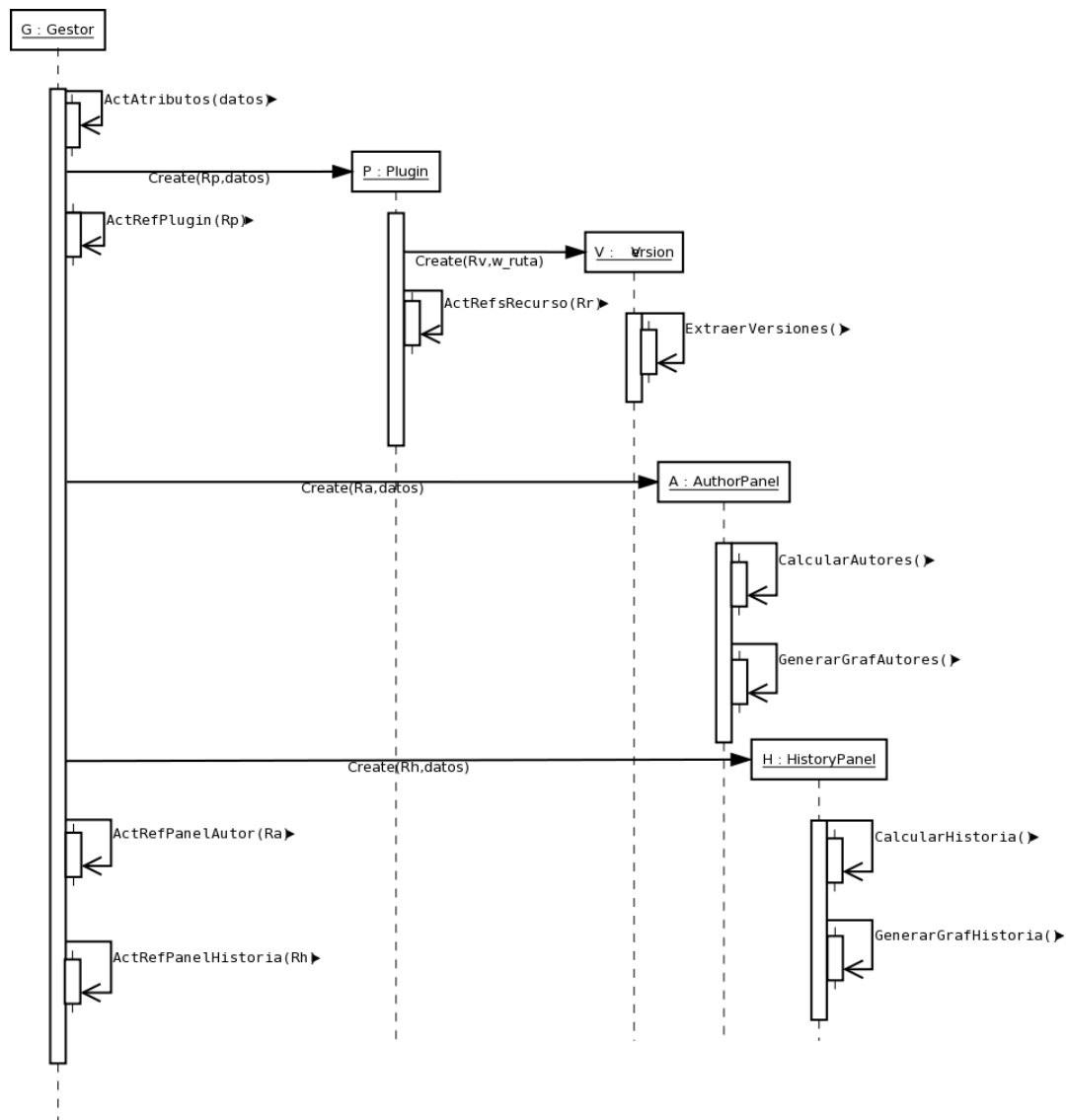
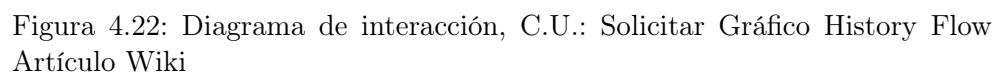


Figura 4.21: Diagrama de interacción, C.U.: Solicitar Gráfico History Flow  
 Artículo Wiki, bloque inferior figura 4.22





A continuación relaciona algunos de los detalles aclaratorios de lo especificado en el diagrama de colaboración de este caso de uso, Figura 4.22.

- El bloque definido como superior de la Figura 4.19 el atributo *datos* corresponde a : (w\_ruta, w\_appWikiId, w\_tituId, w\_modId, w\_opcion, w\_autorId).
- Usaremos *Gestor* para designar la clase o un objeto de la clase *WebHistoryflowMain*.
- Usaremos *Versión* para designar la clase o un objeto de la clase *Version-Builder*.
- El mensaje *ActRefFlowMain()* de la Figura 4.21 , indica la asociación entre *Recurso-Gestor*.
- El mensaje *ActRefPlugi()* de la Figura 4.21 , indica la asociación entre *Gestor-Plugin*, en este caso el plugin de History Flow.
- El mensaje *ActRefPanelAutor()*, *ActRefPanelHistoria()* de la Figura 4.21 , indica la asociación entre *Gestor-AuthorPanel* y *Gestor-HistoryPanel* respectivamente.
- El mensaje *ActRefVersion()* de la Figura 4.21 indica la asociación entre *Plugin-Versión*.
- El mensaje *ExtraerVersiones()* Figura 4.21 indicará la necesidad de proporcionará al objeto *Plugin* los datos obtenidos.
- Los mensajes *CalcularAutores()* y *GenerarGrafAutores()* de la Figura 4.21 indican el cálculo de los distintos autores en las versiones y la generación del gráfico correspondiente.
- Los mensajes *CalcularHistoria()* y *GenerarGrafHistoria()* de la Figura 4.21 indican el cálculo datos implicados en las versiones a lo largo del flujo de la historia del artículo y la generación del gráfico correspondiente.

#### 4.2.1.3. Diagrama de colaboración de CU con formato de salida History Flow fichero en svn

Para este diagrama teniendo en cuenta que solo existen unos pequeños detalles de diferencia respecto del caso de uso para un articulo wiki, se dibujará solo la parte correspondiente al *bloque inferior*. La principal diferencia consiste en la forma de procesar los datos por parte del plugin ya que el plugin de History Flow extrae la versiones a partir del analizador sintáctico SAX de Java, desde una clase a parte y en cambio el plugin de Richard Feard utiliza el analizador DOM y realiza la operación de extracción desde el mismo plugin.

Concretamente Richard utiliza una herramienta Java llamada SVNKit, utilizando un analizador sintáctico DOM. SVNKit es una herramienta Java de código fuente abierto, cumple todas las características de subversión y proporciona API's para trabajar con subversiones, acceder a ellas y manipular los repositorios de subversión, todo dentro de la aplicación Java. Una de las principales diferencias entre SAX y DOM es que el primero realiza una lectura secuencial y una vez leído el dato no hay vuelta atrás, a diferencia de DOM que si lo permite, entre otras cosas. Cuando se describa el apartado de *evaluación y pruebas* se expondrá con mas detalle las características de uno y otro analizador.

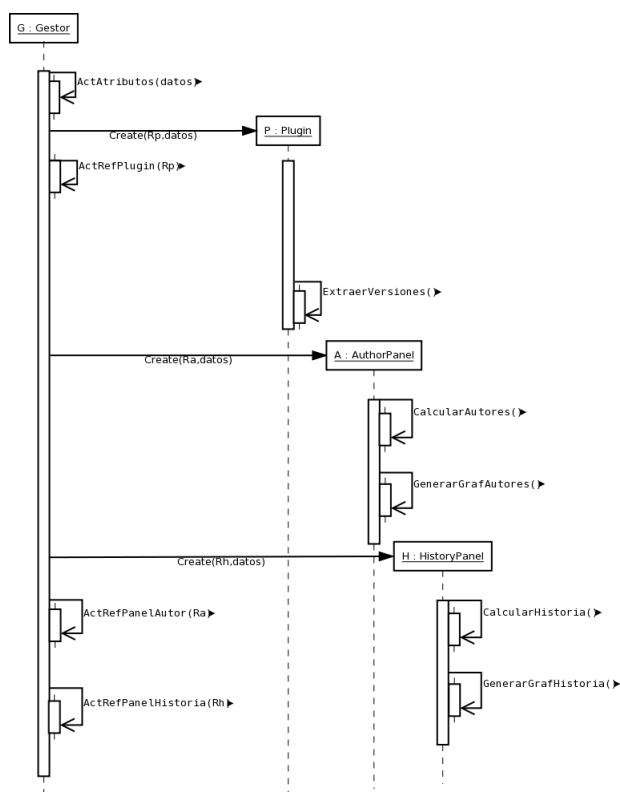


Figura 4.23: Diagrama de interacción, C.U.: Solicitar Gráfico History Flow de un fichero de una svn, bloque inferior.

Algunos de los detalles a mencionar del diagrama de la Figura 4.23 para este caso de uso serian:

- El bloque definido como superior de la Figura 4.19 el atributo *datos* corresponde a : (w\_ruta, w\_opcion, w\_appWikiId, w\_caminoId, w\_tituId, w\_modId, w\_autorId).

- El mensaje *ActRefPlugin()* de la Figura 4.21 , indica la asociación entre *Gestor-Plugin*, en este caso el plugin de Richard Feard.
- Queda eliminada la colaboración con el objeto *V:Versión* ya que es el mismo plugin el que genera la extracción de los datos del fichero de la subversión.
- Aparece el mensaje *ExtraerVersiones()* en el mismo objeto Plugin.

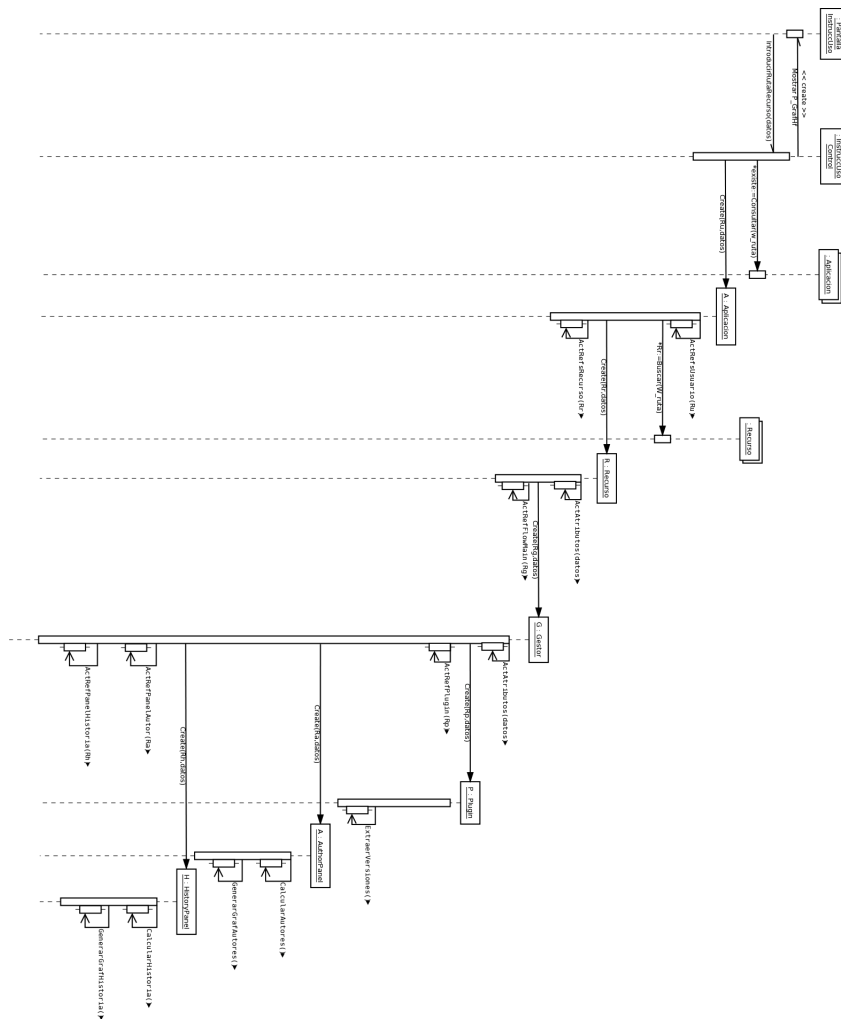


Figura 4.24: Diagrama de interacción, C.U.: Solicitar Gráfico History Flow de un fichero de una svn.

#### 4.2.1.4. Diagrama de colaboración de CU con formato de salida Prefuse artículo wiki

El diagrama de colaboración correspondiente a los casos de uso relacionados con el gráfico de *Prefuse* se representará a partir del objeto *Recurso* por ser la parte inicial similar a los anteriores casos de uso, por lo tanto sólo se representará el diagrama ampliado correspondiente al bloque inferior, Figura 4.25, que es donde se encuentran las diferencias y también se representará otro con todo el conjunto completo, Figura 4.26.

Este diagrama de interacción de la figura 4.26 relacionado con artículos wiki, será en representación de los casos de uso pertenecientes a Prefuse, ya sea desde el punto de vista de creación del gráfico como de la generación del fichero de datos en xml.

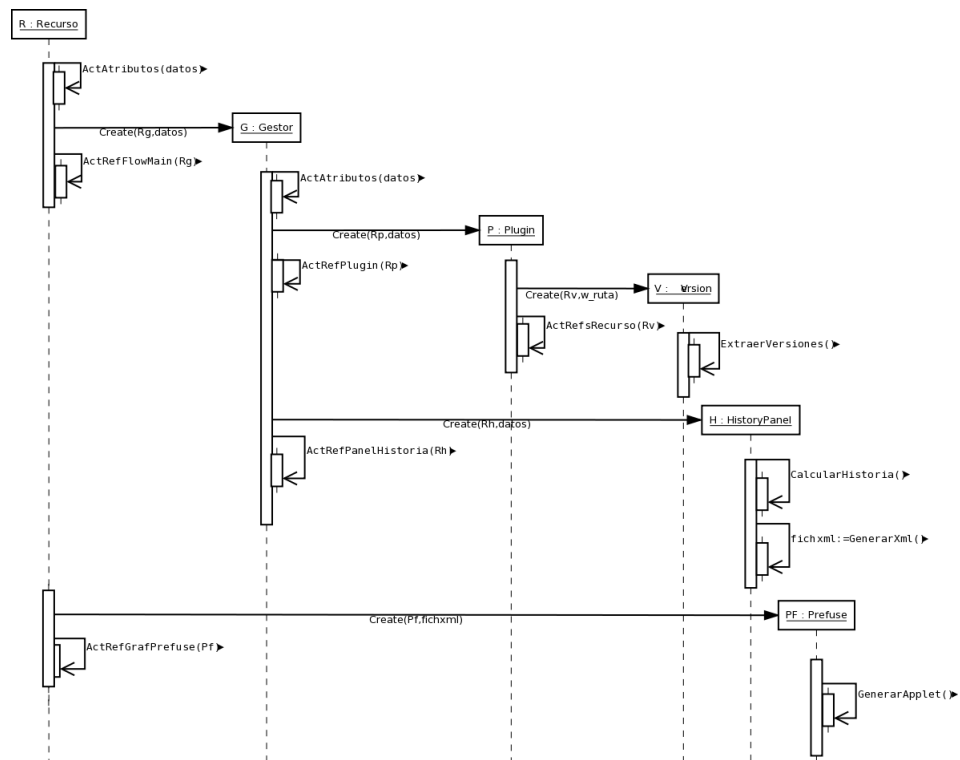


Figura 4.25: Diagrama de interacción, C.U.: Solicitar Gráfico Prefuse de un artículo wiki, bloque inferior.

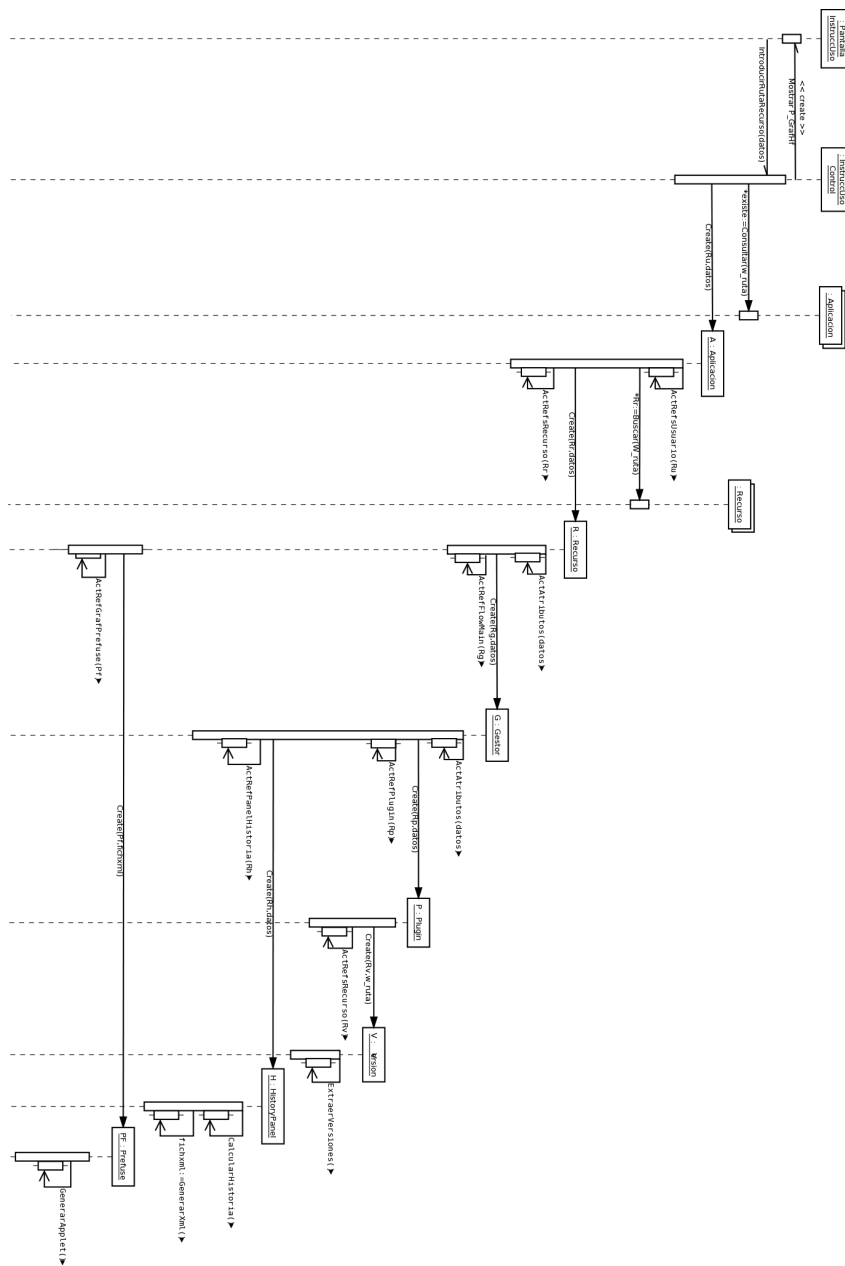


Figura 4.26: Diagrama de interacción, C.U.: Solicitar Gráfico Prefuse de un artículo wiki.

A continuación relaciono algunas aclaraciones y especificaciones del diagrama.

- En la Figura 4.26 el atributo *datos* corresponde a : (w\_ruta, w\_opcion, w\_appWikiId, w\_tituId).
- En la Figura 4.26 usaremos *Prefuse* para designar la clase o un objeto de la clase *GraphView\_HF*.
- El mensaje *ActRefGrafPrefuse()* de la Figura 4.25 , indica la asociación entre *Recurso-Prefuse*.
- El mensaje *fichxml:=GenerarXml()* de la Figura 4.25 , indica la creación de un fichero xml como resultado del procesado de las distintas versiones del artículo wiki.
- El mensaje *GenerarApplet()* de la Figura 4.25 , indica la creación de un *gráfico tipo applet* que genera *nodos y aristas* de la última versión del artículo wiki.

#### 4.2.1.5. Diagrama de colaboración de CU con formato de salida Prefuse fichero en svn

Este diagrama de interacción de la figura 4.28 relacionado con ficheros en svn, será en representación de los casos de uso pertenecientes a Prefuse, ya sea desde el punto de vista de creación del gráfico como de la generación del fichero de datos en xml.

La diferencia a representar con respecto al diagrama del caso de uso anterior es que en este caso de uso no es necesario la creación de una instancia de la clase Versión ya que la recogida de los datos de la versión se realiza en la misma clase Plugin.

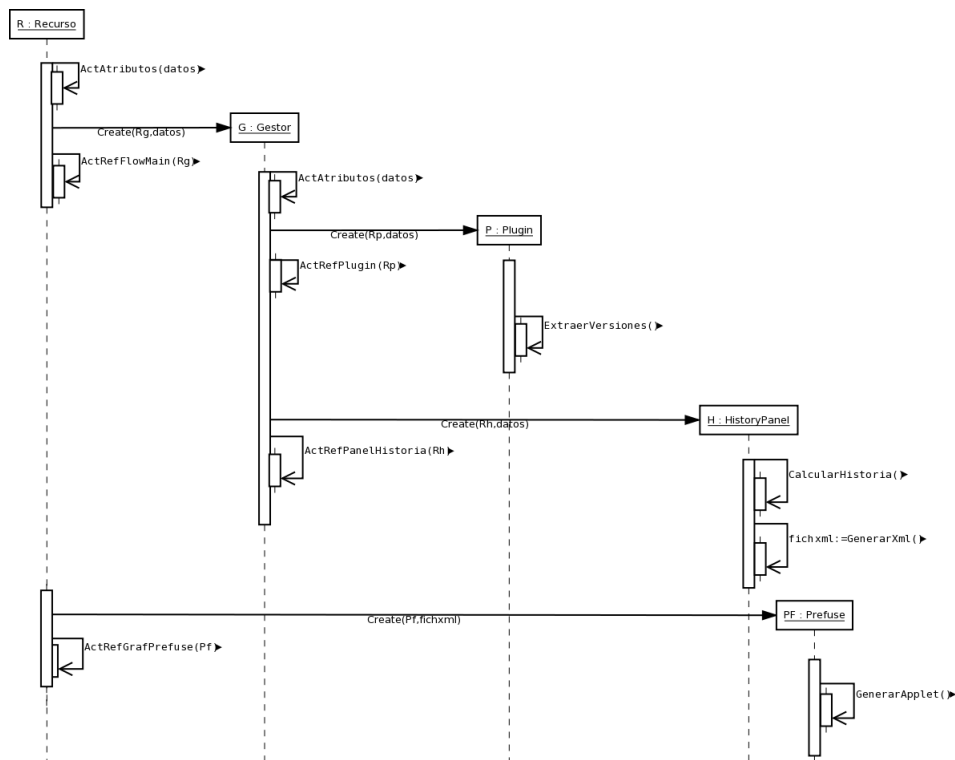


Figura 4.27: Diagrama de interacción, C.U.: Solicitar Gráfico Prefuse de un fichero de una svn, bloque inferior.



Figura 4.28: Diagrama de interacción, C.U.: Solicitar Gráfico Prefuse de un fichero de una svn.



#### 4.2.1.6. Diagrama de colaboración de CU con formato de salida Gephi artículo wiki

Por motivos de claridad a en la visualización de los detalles se representará los diagramas de interacción para los casos de uso relacionado con Gephi en función de artículos wiki. Al igual que en casos anteriores representaré el diagrama de interacción completo y el correspondiente al bloque inferior.

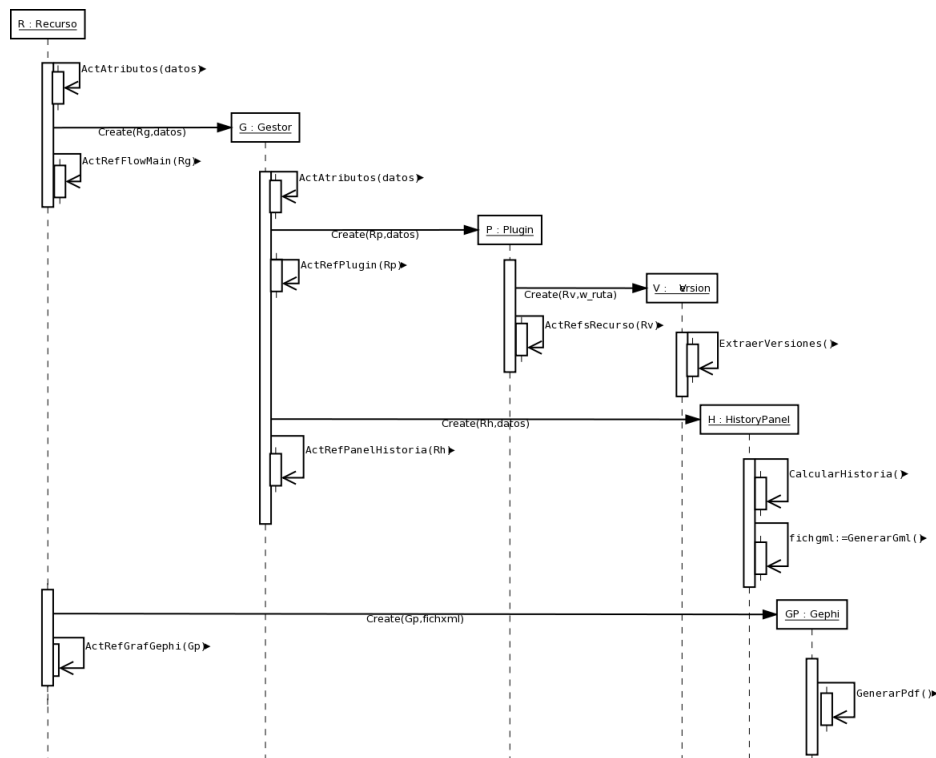
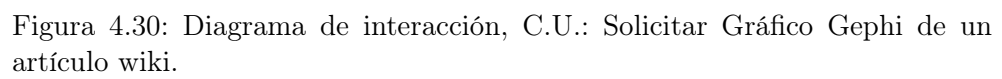


Figura 4.29: Diagrama de interacción, C.U.: Solicitar Gráfico Gephi de un artículo wiki, bloque inferior.



Características a tener en cuenta en este caso de uso relacionado con Gephi, ya sea a nivel de salida de *gráfico* como de *datos gml*.

- En la Figura 4.30 usaremos *Gephi* para designar la clase o un objeto de la clase *RankingGraph*.
- El mensaje *ActRefGrafGephi()* de la Figura 4.29 , indica la asociación entre *Recurso-Gephi*.
- El mensaje *fichgml:=Generargml()* de la Figura 4.29 , indica la creación de un fichero gml como resultado del procesado de las distintas versiones del artículo wiki.
- El mensaje *GenerarPdf()* de la Figura 4.29 , indica la creación de un *pdf* que genera *nodos y aristas* de la última versión del artículo wiki.

#### 4.2.1.7. Diagrama de colaboración de CU con formato de salida Gephi fichero en svn

Igualmente por claridad a en la visualización de los detalles se representará los diagramas de interacción para los casos de uso relacionado con Gephi en función de un fichero perteneciente a una svn. Al igual que en casos anteriores representaré el diagrama de interacción completo y el correspondiente al bloque inferior que son los que muestran variaciones.

También en el caso de uso relacionado con *Prefuse* la diferencia a representar en las svn en Gephi con respecto al diagrama del caso de uso anterior, es que este caso de uso, no necesita la creación de una instancia de la clase *Versión* ya que la recogida de los datos de la versión se realiza en la misma clase *Plugin*.

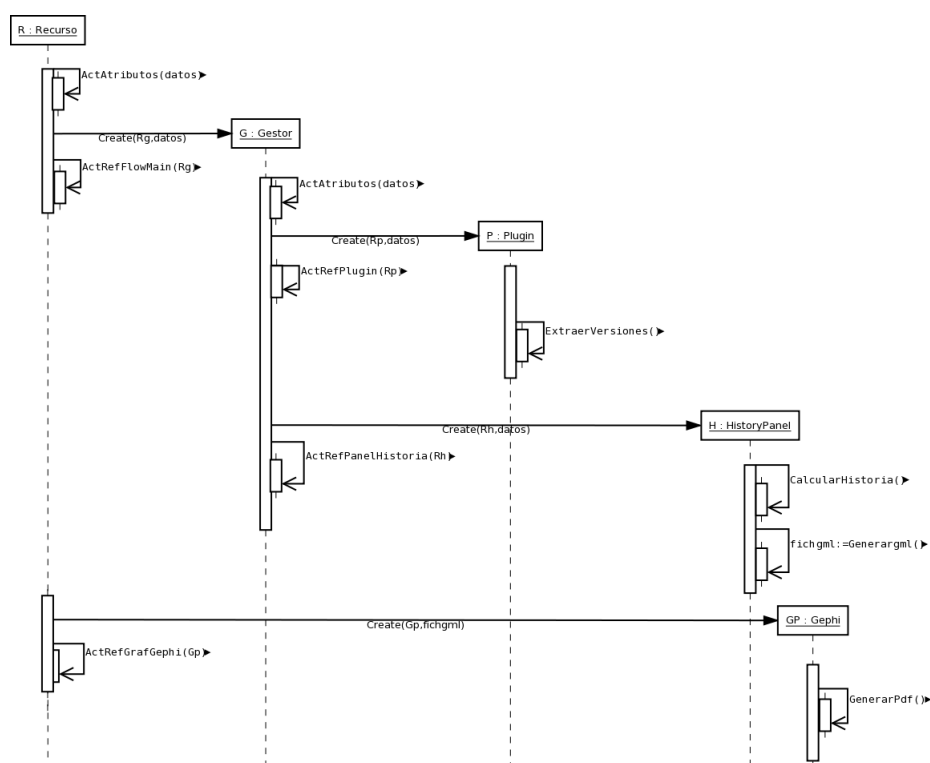


Figura 4.31: Diagrama de interacción, C.U.: Solicitar Gráfico Gephi de un fichero de una svn, bloque inferior.

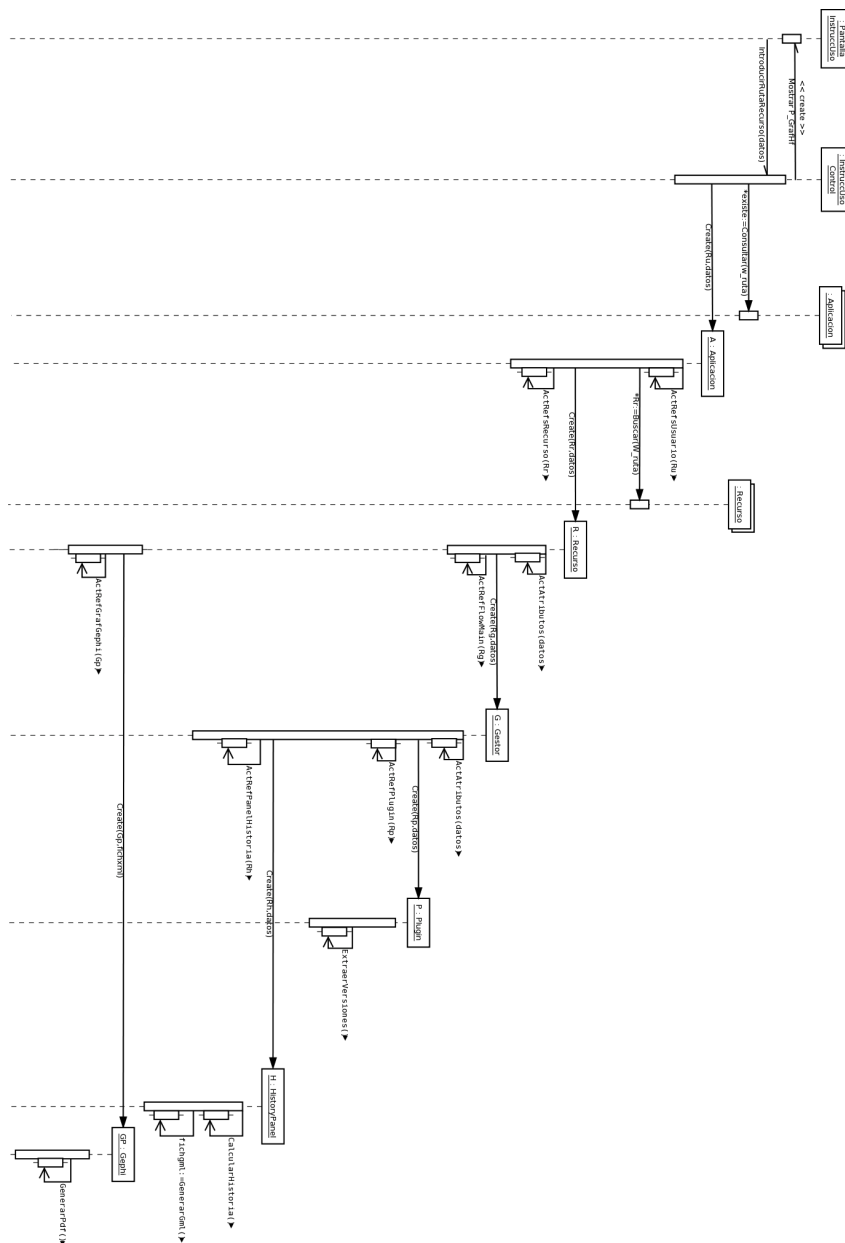
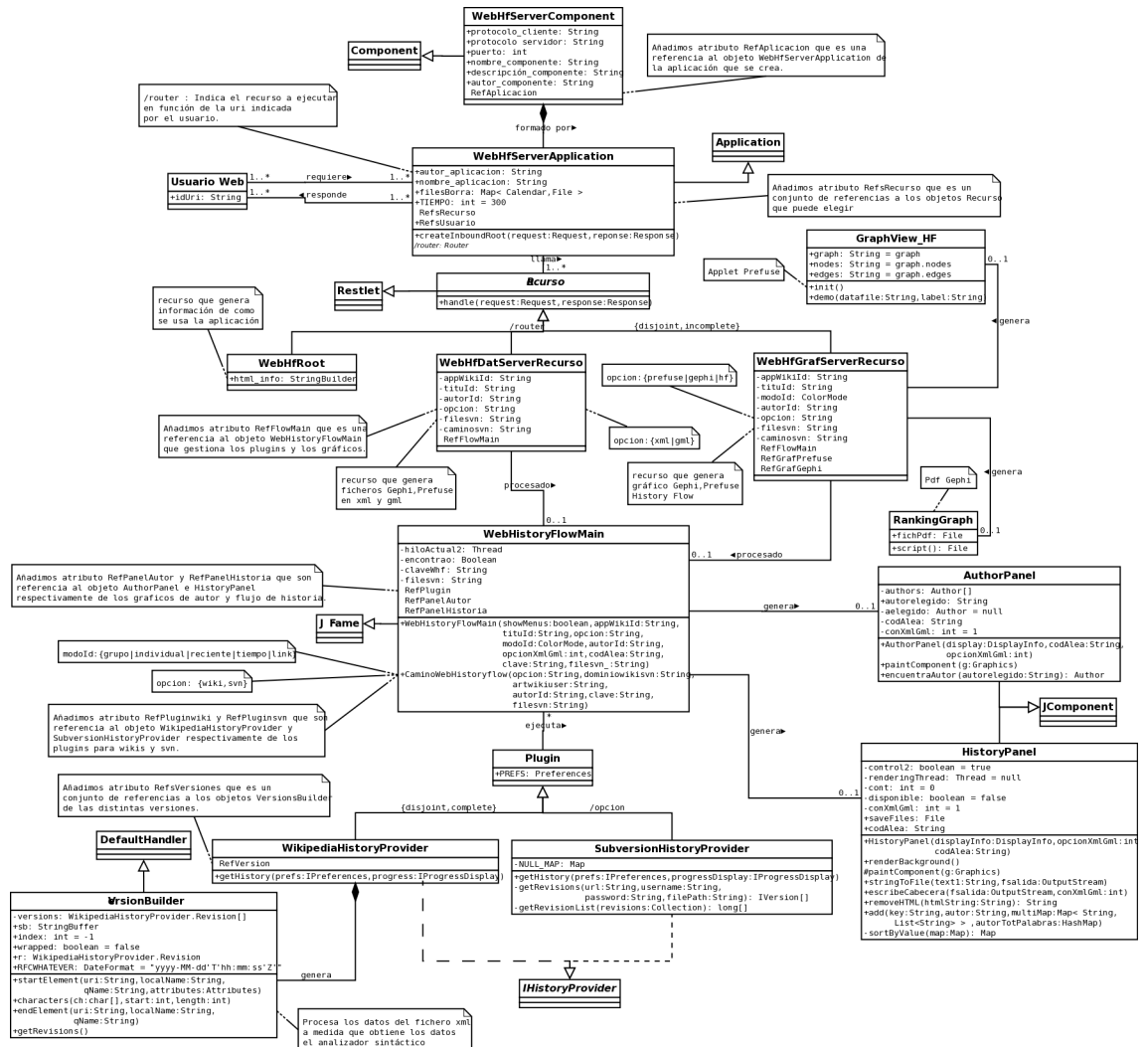


Figura 4.32: Diagrama de interacción, C.U.: Solicitar Gráfico Gephi de un fichero de una svn.

### 4.2.2. Diseño de clases y asociaciones

Con la representación del diagrama de la Figura 4.33 se visualiza detalladamente cual será el diseño de las principales clases que intervienen en el proyecto. Se define el ámbito de los datos (*público, privado*), el tipo de dato (*String, int* etc...) , la *navegabilidad* de las clases y las funciones principales que se usan en cada una de ellas.



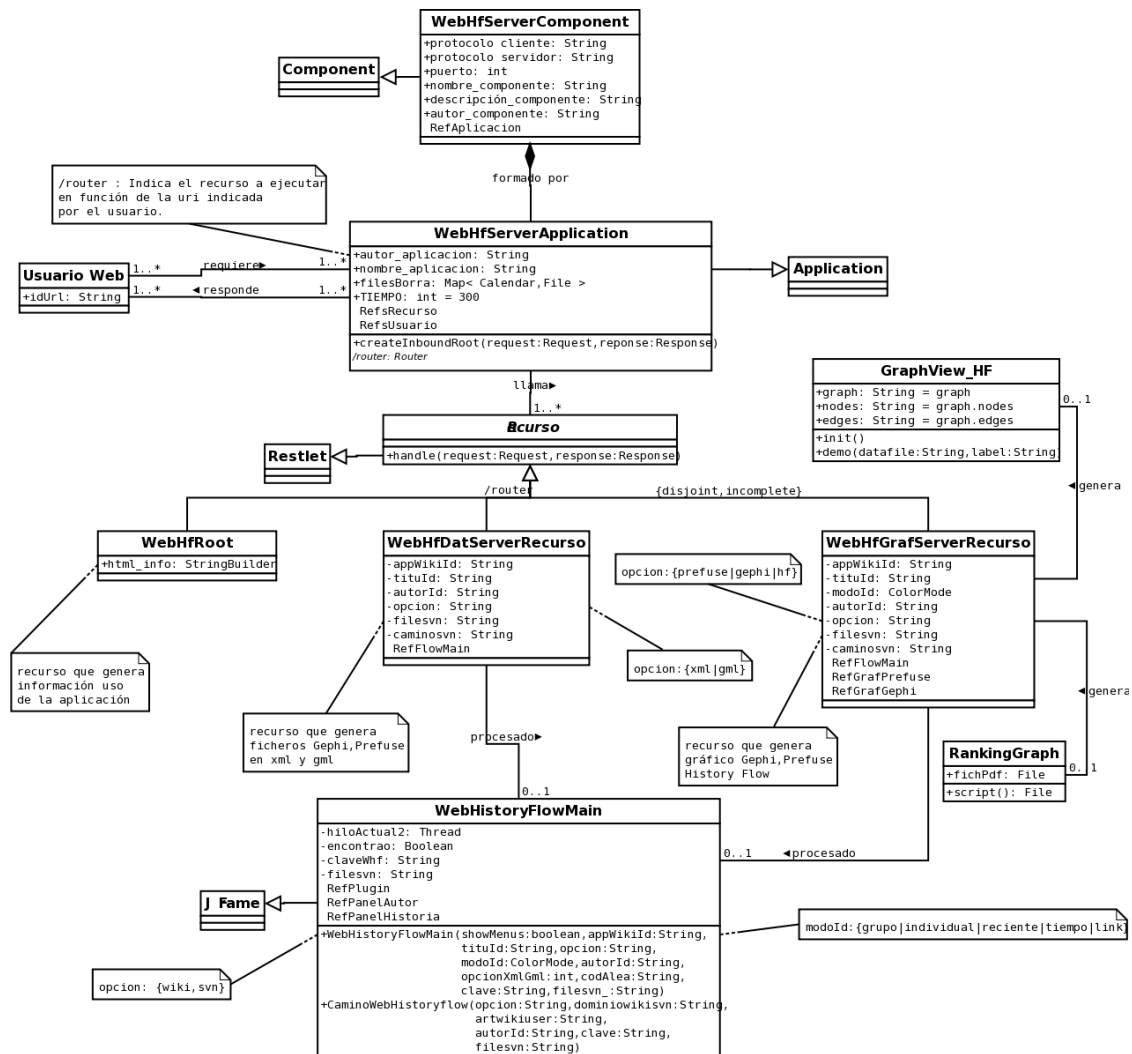


Figura 4.34: Diseño de clases y asociaciones, bloque superior.

Relación de Navegabilidades y atributos añadidos a las Clases de la Figura 4.34.

- WebHfServerComponent —> WebHfServerApplication
- WebHfServerApplication —> Usuario Web
- Usuario Web —> WebHfServerApplication
- WebHfServerApplication —> Recurso
- WebHfRoot —> WebHistoryFlowMain
- WebHfDatServerRecurso —> WebHistoryFlowMain

- WebHfGrafServerRecurso —> WebHistoryFlowMain
- WebHfGrafServerRecurso —> GraphView\_HF (monodireccional)
- WebHfGrafServerRecurso —> RankingGraph (monodireccional)
- Añadimos el atributo *RefsRecurso* que es un conjunto de referencias a los objetos *Recurso* que puede llamar.
- Añadimos el atributo *RefsUsuario* que es un conjunto de referencias a los objetos *Usuario Web* que solicitan los recursos.
- Añadimos el atributo *RefFlowMain* que es una referencia a un objeto *WebHistoryFlowMain* gestor de plugins y gráficos.

**Relación de Navegabilidades y atributos añadidos a las Clases de la Figura 4.35.**

- WebHistoryFlowMain —> AuthorPanel (monodireccional)
- WebHistoryFlowMain —> HistoryPanel (monodireccional)
- WebHistoryFlowMain —> Plugin
- WikipediaHistoryProvider —> VersionBuilder (monodireccional)
- Añadimos el atributo *RefPlugin* que es una referencia a un objeto *Plugin* del artículo wiki o fichero svn a procesar.
- Añadimos el atributo *RefPanelAutor* que es una referencia a un objeto *AuthorPanel* de los gráficos de autores a mostrar.
- Añadimos el atributo *RefPanelHistoria* que es una referencia a un objeto *HistoryPanel* de los flujos gráficos de información.
- Añadimos el atributo *RefVersion* que es una referencia a un objeto *VersionBuilder* de las versiones a generar.



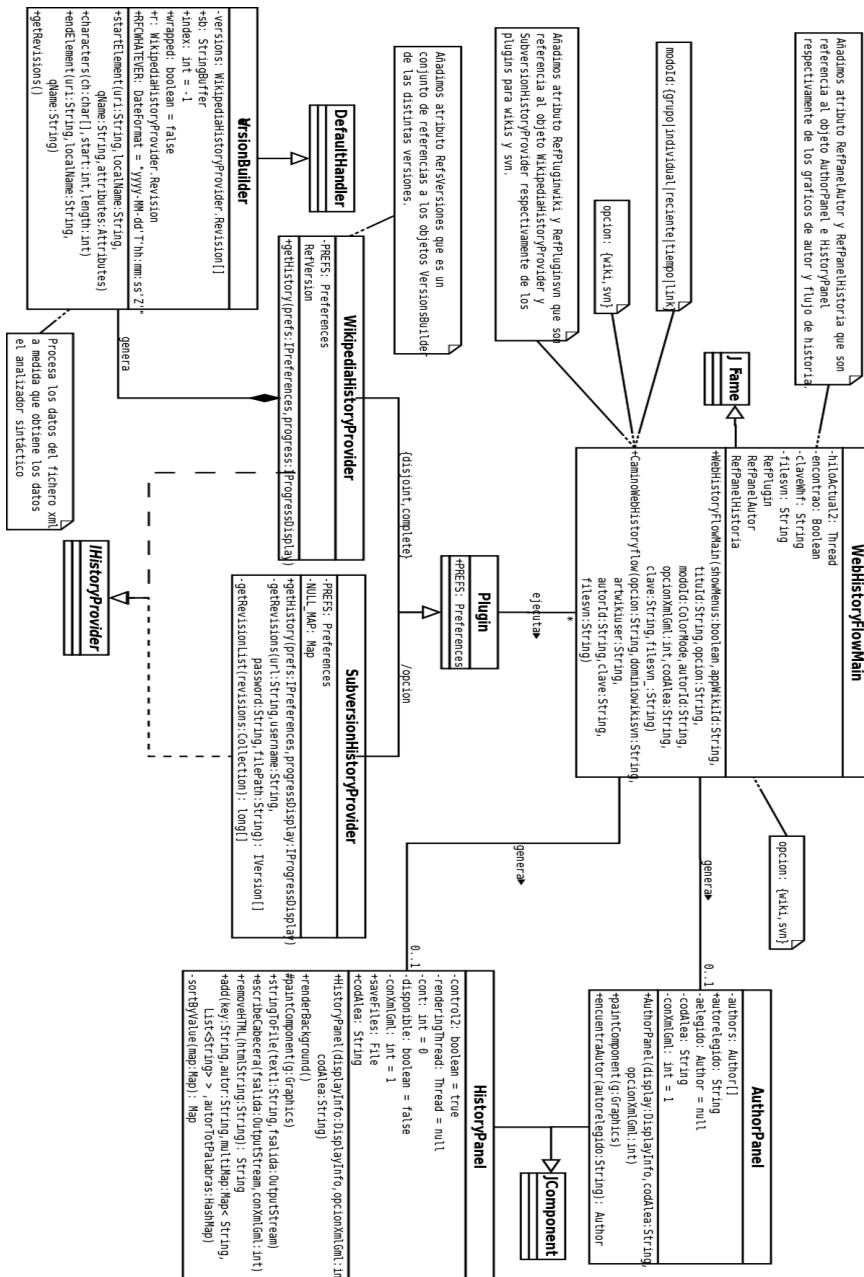


Figura 4.35: Diseño de clases y asociaciones, bloque inferior.

Algunas de las aclaraciones sobre el *diagrama de diseño de las clases* serían las siguientes:

- En la clase raíz, *WebHfServerComponent*, los atributos especificados definen las características de los distintos protocolos a usar ya sean para la comunicación con el cliente, a través del conector de cliente *CLAP* o

con el servidor a través del conector *HTTP*. Así mismo se puede definir el *número de puerto* a usar en caso de ejecuciones en modo local o predisposiciones para futuras ejecuciones en servidores concretos.

- La clase *WebHfServerApplication* es el resultado de la definición de un host virtual por parte de la clase raíz, a partir de la cual se podrá expandir la implementación del proyecto bajo el encapsulamiento del *framework de Restlet*.
- La clase *WebHfServerApplication* se encarga de establecer el enrutado de los distintos recursos disponibles para el cliente, recogiendo el contexto del requerimiento solicitado por el cliente en cada momento y estableciendo un hilo con cada uno de ellos.
- El método *CreateInboundRoot()*, es el que se llama al inicio de una *clase aplicación* en el framework de Restlet, va a ser el encargado de crear el Restlet raíz y crea una ruta raíz para enviar llamadas a los recursos del servidor.
- Cada clase que hereda de la clase *Recurso*, que a su vez, ésta también hereda de la clase *Restlet*, incluyen el método *handle()* que es un manejador de las peticiones realizadas por el cliente.
- Las clases como *WebFfServerApplication* o *Recurso* de la Figura 4.34, que heredan de las clases *Application* y *Restlet* respectivamente son clases donde la ejecución de las mismas se pueden realizar en modo concurrente.
- A la hora de instanciar un objeto de la clase *WebHistoryFlowMain* de la Figura 4.35 el orden de ejecución, ya sean datos o gráficos, sería el siguiente :
  1. Ejecutar el plugin que corresponda para extraer las versiones contenidas, ya sean de un artículo wiki o de un fichero de una svn ( clases: *WikipediaHistoryProvider*, *SubversionHistoryProvider* ).
  2. Calcular los datos y gráficos correspondientes a los autores y flujos de historia de información ( clases: *AuthorPanel*, *HistoryPanel* ).
  3. Visualización del gráfico correspondiente a History Flow, Gephi o Prefuse ( clases: *GraphView\_HF*, *RankingGraph* ).
- A través del constructor de la clase *WebHistoryflowMain* se realizan las llamadas para el cálculo de los datos relacionados con los autores y el flujo de estos en las distintas versiones.
- El método *CaminoWebHistoryflow()* de la clase *WebHistoryflowMain* envía el dominio y el nombre del artículo, ejem.: (es.wikipedia.org , UCA) para obtener los datos de una url determinada ya sea de una wiki o de una

svn. Este método es el encargado de instanciar los plugins (historyflow - Richard Feard) para wikis y ficheros de subversiones respectivamente.

### 4.3. Implementación

Una vez realizado todo el estudio de análisis y diseño de la aplicación es el momento de pasar a la implementación, para ello he dividido el trabajo a realizar en varios bloques ya que los temas y apartados a tratar son bastante extensos.

- Características de la implementación.
- Trabajar con el *framework de Restlet* para encapsular la aplicación History Flow.
- Características comunes para recursos History Flow, Prefuse y gephi.
- Modificación de la *aplicación History Flow*, eliminando partes no útiles en un entorno web, como puede ser todo lo relacionado con la interfaz de usuario y agregando las clases y métodos que vayan haciendo falta.
- Implementar Prefuse y Gephi en WebHistoryflow.

#### 4.3.1. Características de la implementación

Para la realización del proyecto se ha utilizado el lenguaje de programación *Java*. Java es el lenguaje que mejor se ha adaptado a las exigencias previstas del proyecto ya que junto con PHP es uno de los mas utilizados para utilización en entornos web. También hay que tener en cuenta que se necesitaba un lenguaje que fuese utilizados por distintas API's que manejaran entornos gráficos y Java se ajustaba a ello.

De entre las distintas API's que usaban Java a la hora de implementar sus aplicaciones o demos encontré la API de Prefuse y la de Gephi ambas con un desarrollo de sus aplicaciones en lenguaje Java y con una implementación bastante sencilla, así que eran ideales para incluirlas como complementos de pruebas en la utilización de WebHistoryFlow en colaboración con otras aplicaciones. En los siguientes párrafos se describirá con más detalle las características mas importantes en sus implementaciones.

Para implementar la aplicación, he teniendo en cuenta que se va a basar en un suministro de recursos a partir de HTTP <sup>1</sup>, así que he alojado WebHistoryFlow en un Servidor Web en este caso Apache y para la ejecución de todo lo relacionado con Java, como puede ser JSP, Servlets y Restlet he optado

---

<sup>1</sup>HTTP es un Protocolo de Transferencia de Hipertexto, la información transmitida sería los recurso y se identifica mediante un localizador uniforme de recursos URL.

por Tomcat, que curiosamente es mantenido y desarrollado por el equipo de Apache.

Inicialmente para evitar pérdidas de tiempo en la implementación y la realización de pruebas, a medida que se estaba desarrollando el proyecto, la ejecución de lo implementado se ha ido probando a través de *Tomcat* en modo local, reduciendo enormemente el tiempo en todas las pruebas realizadas, posteriormente cuando ha estado operativa la versión definitiva se ha pasado a un servidor privado como *spilin.uca.es:9080* a través de puerto 9080 como puede verse en este caso.

Unas de las herramientas usadas para la realización del proyecto, el IDE Netbeans 7.0, ha sido de muchísima utilidad ya que a parte de realizar en él, la implementación y depuración de la aplicación, he podido aprovechar la posibilidad de incorporarle un servidor web como Tomcat y así ejecutar sobre la marcha las implementaciones que se iban realizando.

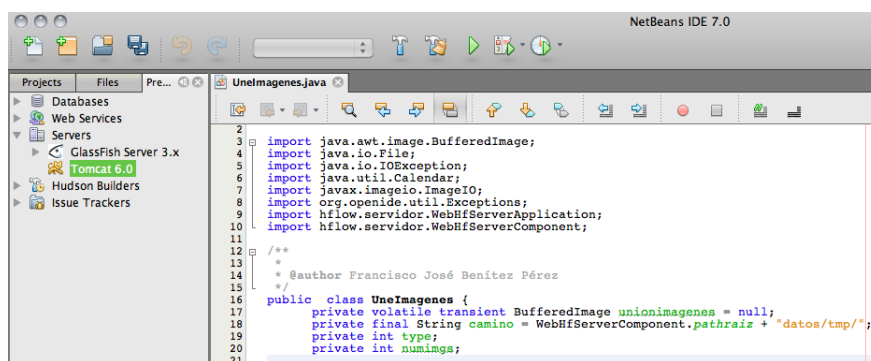
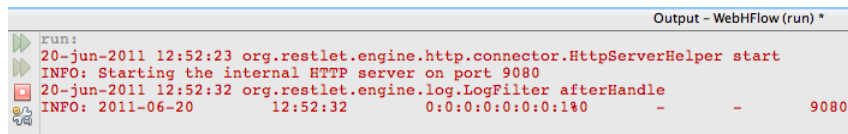


Figura 4.36: IDE, Interfaz de Desarrollo Integrado, Netbeans 7.0

Aunque inicialmente, para la implementación de los primeros bloques de código estuve trabajando con el servidor acoplado a Netbeans Tomcat, cuando se entro en el apartado de Restlet, que ha sido el Framework para encapsular la aplicación WebHistroyFlow, vi que te proporcionaba la posibilidad de ejecutar al inicio de la aplicación desde las mismas líneas de código, un servidor web virtual incluyendo lógicamente el puerto para su implementación. Este modo de trabajar ofrecía la posibilidad de ejecutar cualquier desarrollo de la aplicación que se estuviera realizando sin la necesidad de instalaciones añadidas, contando implícitamente con un servidor web que ejecutaba todo el código Java. Como puede verse en la Figura 4.37 cuando se inicia la aplicación se ejecuta inicialmente el servidor virtual en el puerto local 9080 y en la siguiente línea información sobre el recurso suministrado desde una petición del cliente.

En cuanto al equipo hardware, he utilizado un *mac*, concretamente un *MacBookPro* (portátil) que a pesar de tener varios años sigue funcionando y



```
run:
20-jun-2011 12:52:23 org.restlet.engine.http.connector.HttpServerHelper start
INFO: Starting the internal HTTP server on port 9080
20-jun-2011 12:52:32 org.restlet.engine.log.LogFilter afterHandle
INFO: 2011-06-20      12:52:32      0:0:0:0:0:0:1*0      -      -      9080
```

Figura 4.37: Ejecución del Servidor Virtual de Restlet.

ejecutando el software de manera eficiente y rápida. Sus características principales son las siguientes:

- Sistema operativo Mac OS X ver.10.6.7
- Procesador 2.33 GHz Intel Core 2 Duo
- Memoria 2 GB 667 M DDR2 SDRAM
- S.O. Multifunción y Multitarea.

#### 4.3.2. Trabajar con el framework Restlet

Bueno el trabajo con el framework de Restlet ha sido uno de los puntos claves para el encapsulado de la aplicación, ya que cuando estuve realizando la fase de búsqueda de información, el método de petición de recursos desde la URI se hacía un poco engorroso. Después de ver que para realizar la petición de un recurso desde la URL había que escribir la biblia, opté por buscar otras opciones tras pedir consejo al tutor del proyecto. Llegamos a la conclusión que Restlet cumplía las expectativas y una vez analizado con qué facilidad podía enrutar Restlet un recurso, me puse manos a la obra.

A continuación expondré algunas de las características de Restlet usadas durante el proyecto así como una pequeña información a título orientativo de este Framework.

##### 4.3.2.1. Introducción a Restlet

Restlet es un framework para la web de código abierto para desarrolladores de Java. Ofrece capacidades completas REST <sup>2</sup> y un estilo de arquitectura nativa para la Web. Restlet es un framework orientado a objetos que proporciona un amplio conjunto de clases y rutinas.

El Marco Restlet apoya todas las formas de la Web de una manera sencilla y unificada. Se puede utilizar tanto para proporcionar como para consumir recursos de la Web, incluyendo páginas y servicios web.

---

<sup>2</sup>REST, técnica de arquitectura software para sistemas hipermedia distribuidos, como la World Wide Web.

Una de las ventajas que tiene Restlet, es que trabaja tanto del lado del cliente como del lado del servidor todas las características de HTTP, como pueden ser, métodos condicionales, negociación de contenido, variación del contenido, compresión de contenido o verificación de contenido.

REST fue creado por Roy T. Fielding, principal arquitecto de HTTP 1.1, que es la columna vertebral del protocolo utilizado en la Web. REST es el acrónimo de Transferencia de Estado Representacional. Se trata de un conjunto de principios que, correctamente aplicado, ayudan a la construcción de arquitecturas software y aplicaciones que se benefician de todas las cualidades de la Web, como puede ser la escalabilidad, uso eficiente de la red y evolución independiente de los Clientes y Servidores.

#### 4.3.2.2. Gestión de Recursos

Los recursos podríamos decir que son los bloques a consumir o proporcionar dentro de la web, es decir puede ser cualquier cosa de interés que sea expuesta por una aplicación y a su vez pueda ser utilizada por otra para su consumo. En nuestro caso estamos hablando de una serie de representaciones gráficas y de unos ficheros de datos que son puestos a disposición de los clientes para su consumo.

Una característica importante de los recursos es que están unidos entre sí, por ejemplo, a través de hipervínculos en documentos HTML o a partir de referencias URI <sup>3</sup> como es en nuestro caso.

Como puede verse en la Figura 4.38, un recurso, una vez identificado, podrá exponer su estado a través de representaciones, conteniendo éstas, metadatos como (tipo soporte, tamaño, conjunto de caracteres) y contenido actual que posee como (imagen binaria, documento de texto). Por ejemplo la representación de la confirmación de un artículo wiki podría ser un fichero en formato XML para su uso con Prefuse, o la representación de la confirmación de un autor perteneciente a un fichero de una svn podría ser una secuencia binaria de JPEG.



Figura 4.38: Unión entre un Identificador, un Recurso y una Representación.

HTTP es el protocolo principal para manipular los recursos en la Web. Se trata de un protocolo de red cliente-servidor: una aplicación cliente, ya sea un

---

<sup>3</sup>URI, es un identificador uniforme de recursos, permitiendo a cualquier usuario, localizar un recurso a partir de su identificador único.

navegador web o cualquier otro tipo de programa, envía una solicitud a una aplicación de un servidor, la procesa y devuelve una respuesta. Esta respuesta puede ser un documento HTML en el caso de un sitio web, un documento XML o cualquier otro tipo de datos.

HTTP no se limita a definir un modelo de mensaje (petición-respuesta como GET), también puede realizar otras operaciones dirigidas a recursos identificados por una URI, (como PUT, POST, OPTION, DELETE, HEAD). *GET* se podría expresar como: Dame una representación del estado actual de un determinado recurso o crea o modifica un determinado recurso con la representación que te envío (*PUT*) o elimina el recurso que te indico (*DELETE*). En la realización de este proyecto, aunque sí se podrían haber utilizados opciones de recurso como PUT por ejemplo para la adicción de nuevos servidores base para la elección de artículos o ficheros, solo se ha utilizado la opción GET, donde a partir de una petición se proporciona un recurso.

#### 4.3.2.3. Componentes y conectores

Los recursos en la web por directamente no pueden ser manipulados por los clientes, necesitan ser distribuidos, accedidos y gestionados eficientemente, para ello están los componentes. Hay muchos tipos de componentes, tales como los agentes de usuario, los servidores de origen, pasarelas o proxies. Los agentes de usuario son los navegadores web (como *Mozilla Firefox*) y los servidores de origen mas comunes son los servidores y motores web (como *Microsoft IIS* y *Apache Tomcat*). El framework de Restlet proporciona todas las tecnología necesaria para desarrollar dichos componentes.

Como puede verse en la Figura 4.39 se muestra la conexión entre dos componentes a través de las dos líneas que representan protocolos de transferencia HTTP o una comunicación TCP/IP, los mensajes de petición y respuesta, y los roles de cliente-servidor.

En REST, las comunicaciones son apátridas, es decir, el cliente envía una solicitud para que sea autosuficientes y el servidor devuelve una respuesta, y luego la comunicación finaliza. Hay muchos tipos de conectores, por ejemplo los conectores de cliente, los conectores de servidores. Los conectores más comunes son los conectores de cliente (como la librería cliente HTTP de Apache), que envían solicitudes y los conectores del servidor (como la librería servidor HTTP de Jetty) que escucha la información entrante de solicitudes y las respuestas a devolver. El Framework de Restlet también proporciona conectores para varios protocolos como HTTP, POP3, SMTP, FILE, CLAP, RIAP y algunos mas.

#### 4.3.2.4. La clase raíz Restlet

El framework de Restlet aunque parte de la superclase Object ya que es Java realmente y es la clase desde la que heredan todas las clases en Java,

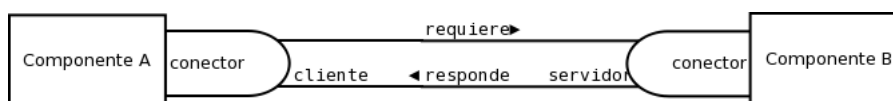


Figura 4.39: Comunicación entre componentes.

tiene como clase de partida la clase *Restlet*, Figura 4.40. La clase *Restlet* implementa una interfaz uniforme y su equivalente sería el paquete de Java *javax.servlet.http.HttpServlet*, y su método principal es *handle()*, que es el encargado de manejar las peticiones y respuestas, además esta clase incorpora los principios de interfaz uniforme para los recursos REST.

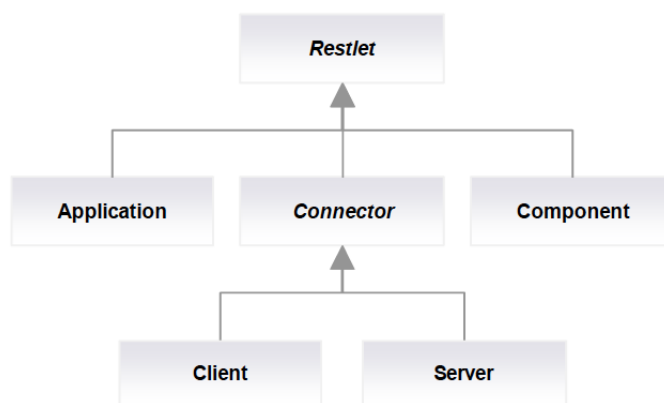


Figura 4.40: Herencia de clases del paquete raíz.

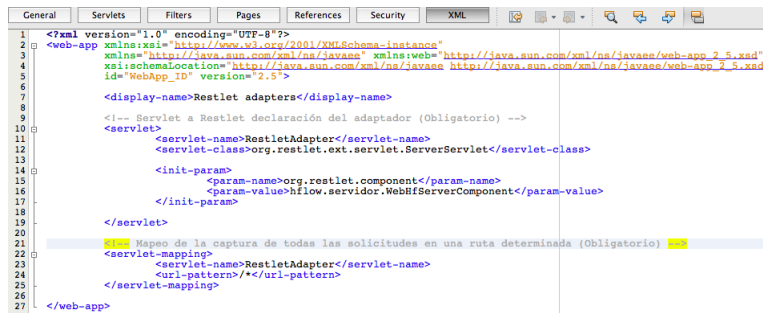
Una de las diferencias mas importantes de *Restlet* frente a los *Servlet* son la abstracción en cuanto a las solicitudes y respuestas ( *Request*, *Response* ), referidas a detalles de bajo nivel en el protocolo HTTP, donde la API de *Servlet* deja bastante trabajo en manos del desarrollador. Un buen ejemplo de esto es que en *Restlet* no hay que analizar manualmente las cabeceras HTTP, ese trabajo ya lo realiza el framework de *Restlet* y la información útil queda disponible en clases de Java y propiedades.

#### 4.3.2.5. Fichero *web.xml* para *Restlet*

Para poder ejecutar la aplicación hecha con el framework de *Restlet* en un entorno web, es necesaria una configuración previa del fichero *web.xml* . Desde este fichero se especifica si la aplicación va a ejecutarse desde una clase tipo *aplicación* o desde una clase tipo *componente*.

A la hora de crear una aplicación podríamos partir en *Restlet* desde una *Aplicación* o desde un *Componente*. Se podría decir que la Clase *Componente*





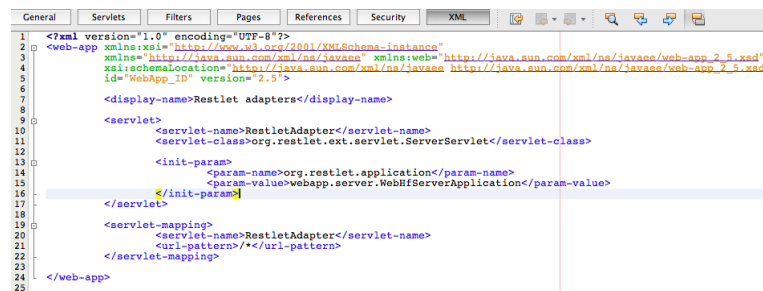
```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3         xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
4         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
5         id="WebApp_ID" version="2.5">
6
7     <display-name>Restlet adapters</display-name>
8
9     <!-- Servlet a Restlet declaración del adaptador (Obligatorio) -->
10    <servlet>
11        <servlet-name>RestletAdapter</servlet-name>
12        <servlet-class>org.restlet.ext.servlet.ServerServlet</servlet-class>
13
14        <init-param>
15            <param-name>org.restlet.component</param-name>
16            <param-value>hflow.servidor.WebHfServerComponent</param-value>
17        </init-param>
18    </servlet>
19
20    <!-- Mapeo de la captura de todas las solicitudes en una ruta determinada (Obligatorio) -->
21    <servlet-mapping>
22        <servlet-name>RestletAdapter</servlet-name>
23        <url-pattern>/*</url-pattern>
24    </servlet-mapping>
25
26 </web-app>

```

Figura 4.41: Fichero web.xml, usando clase Component.

es una especie de cajón dónde podemos meter una o mas Clases Aplicación. Entiendo que si se quiere partir de una raíz que quiera manejar distintos tipos de aplicaciones, tomando por ejemplo como aplicación, distintas tecnologías, (como restlet, servlet, jsp ...) lo mas adecuado es crear una clase Componente y a partir de ella generar las correspondientes clases Aplicación. Cada una de estas aplicaciones a su vez tendría agrupados los distintos recursos a consumir.



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3         xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
4         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
5         id="WebApp_ID" version="2.5">
6
7     <display-name>Restlet adapters</display-name>
8
9     <servlet>
10        <servlet-name>RestletAdapter</servlet-name>
11        <servlet-class>org.restlet.ext.servlet.ServerServlet</servlet-class>
12
13        <init-param>
14            <param-name>org.restlet.application</param-name>
15            <param-value>webapp.server.WebHfServerApplication</param-value>
16        </init-param>
17    </servlet>
18
19    <servlet-mapping>
20        <servlet-name>RestletAdapter</servlet-name>
21        <url-pattern>/*</url-pattern>
22    </servlet-mapping>
23
24 </web-app>

```

Figura 4.42: Fichero web.xml, usando clase Application.

Por lo tanto para permitir la extensibilidad de la aplicación Webhistoryflow la implementé de manera que fuese posible acoplar cualquier otra aplicación, opté por comenzar desde una clase Componente y agrupar todos los recurso de gráficos y ficheros xml,gml en una clase Aplicación. Aunque como ya comenté antes en este caso también podría haber arrancado desde la clase Aplicación. En las figura 4.41 - 4.42 podemos ver configuraciones para arrancar desde un Componente y desde una Aplicación.

#### 4.3.2.6. Implementar las rutas en Restlet

Una de las ventajas de usar el framework de Restlet es la facilidad para acceder a los distintos recursos que proporcionan las aplicaciones.

Si se hubiese utilizado el paquete de Java javax.servlet.http.HttpServlet para la gestión de peticiones a través del protocolo HTTP, nos hubiésemos

encontrado una petición de la siguiente forma en la URL del explorador web:

```
http://localhost:8111/servlet/myservlet?wikiId=wiki/  
Es.wikipedia.orgEs.art=UCAEs.modId=grupo
```

Con la utilización de Restlet la misma petición se vería de la siguiente forma :

```
http://localhost:8111/wiki/es.wikipedia.org/UCA/grupo
```

Como podéis observar la forma de realizar la petición es muchísimo mas clara y es el propio framework de Restlet el que se encarga a través de rutas previamente establecidas a la hora de implementar el código, de llamar al recurso correspondiente. Si la llamada a realizar hubiese sido a un recurso con algunos parámetros mas, se hubiese hecho eterno el recuso a solicitar desde un explorador por ejemplo. Desde Restlet con dos simples instrucciones de código se le indica que recoja el contexto de la operación y que enrute el recurso correspondiente:

1. Router router = new Router(getContext());
2. router.attach("/wiki/appWikiId/tituId/modoId", new WebHfWikiServerRecurso());

Lo que hace Restlet es ver si la petición URI realizada se corresponde con la ruta previamente definida en alguna de las aplicaciones. En este caso, sí coincide *localhost:8111/wiki/es.wikipedia.org/UCA/grupo* con la ruta definida, es decir:

1. Después de la dirección raíz, *localhost:8111*, del servidor hay cuatro parámetros divididos por una barra.
2. El primer parámetro *wiki*, coincide.
3. Los otros tres parámetros coinciden con la cantidad de parámetros pendientes de recibir para coincidir con esta ruta.

En este caso de definición de una ruta podríamos decir que *wiki* es comprobado como constante y las demás que están encerradas entre llaves serías variables ya que cualquiera de las tres pueden tomar distintos tipos de valores. Estas variables serían procesadas posteriormente en el recurso que se solicita. Estos tres parámetros indicados además de indicar una ruta definen tres valores a utilizar en el recurso.

```

/*
 * @author Francisco José Benítez Pérez
 */
public class WebHfServerApplication extends Application {
    public static final Map< Calendar, File > filesBorra = new HashMap< Calendar, File >(); // ficheros a borrar rel. fecha-fichero
    public static final int TIEMPO = 300; // define el tiempo en segundos para borrado de ficheros.

    /**
     * CreateInboundRoot() crea una ruta raíz para enviar llamadas a
     * los recursos del servidor.
     * @return
     */
    @Override
    public synchronized Restlet createInboundRoot(){
        Router router = new Router(getContext());

        // Muestras información en el recurso Raíz, formas de usar webHistoryFlow
        router.attach("/", new WebHfRoot());

        // muestra gráfico de una wiki en modo HistoriFlow (SIN autor) INICIAL
        router.attach("/wiki/{appwikiId}/{tituId}/{modoId}",
            new WebHfWikiServerRecurso());

        // muestra fichero .xml para la API de Prefuse, en función de una wiki.
        router.attach("/svn/prefuse/{appwikiId}/{caminoId}/{tituId}/xml",
            new WebHfSvnPrefuseServerRecursoXml());

        // muestra fichero .xml para la API de Prefuse, en función de una wiki.
        router.attach("/wiki/prefuse/{appwikiId}/{tituId}/xml",
            new WebHfWikiPrefuseServerRecursoXml());

        // muestra fichero .gml para la API Gephi, en función de una svn.
        router.attach("/svn/gephi/{appwikiId}/{caminoId}/{tituId}/gml",
            new WebHfSvnGephiServerRecursoGml());

        // muestra fichero .gml para la API Gephi, en función de una wiki.
        router.attach("/wiki/gephi/{appwikiId}/{tituId}/gml",
            new WebHfWikiGephiServerRecursoGml());

        // muestra gráfico para la API Gephi en formato PDF, en función de una Svn.
        router.attach("/svn/gephi/{appwikiId}/{caminoId}/{tituId}",
            new WebHfSvnGephiServerRecurso());

        // muestra gráfico para la API Gephi en formato PDF, en función de una wiki.
        router.attach("/wiki/gephi/{appwikiId}/{tituId}",
            new WebHfWikiGephiServerRecurso());

        // muestra gráfico de una wiki en modo Prefuse (SIN indicar opcion individual=autor)
        router.attach("/wiki/prefuse/{appwikiId}/{tituId}/",
            new WebHfWikiPrefuseServerRecurso());

        // muestra gráfico de una svn en modo Prefuse (SIN indicar opcion individual=autor)
        router.attach("/svn/prefuse/{appwikiId}/{caminoId}/{tituId}/",
            new WebHfSvnPrefuseServerRecurso());

        // muestra gráfico de una svn en modo HistoriFlow (SIN autor)
        router.attach("/svn/{appwikiId}/{caminoId}/{tituId}/{modoId}",
            new WebHfSvnServerRecurso());

        // muestra gráfico de una wiki en modo HistoriFlow (CON autor)
        router.attach("/wiki/{appwikiId}/{tituId}/{modoId}/{autorId}",
            new WebHfWikiAutorServerRecurso());

        // muestra gráfico de una svn en modo HistoriFlow (CON autor)
        router.attach("/svn/{appwikiId}/{caminoId}/{tituId}/{modoId}/{autorId}",
            new WebHfSvnAutorServerRecurso());

        /* crea una ruta a un directorio, en este caso al de grsfprefuse
         * imprescindible para poder ejecutar el applet desde restlet
         */
        router.attach("/wiki/prefuse/{appwikiId}/{tituId}", htmlApplet);
        router.attach("/svn/prefuse/{appwikiId}/{caminoId}/{tituId}", htmlApplet);

        return router;
    }
}

```

Figura 4.43: Clase Application para WebHistoryFlow, agrupa recursos.

Como se muestra en la Figura 4.43 la clase WebHfServerApplication extiende la clase Application que es la encargada de agrupar todas las posibles rutas que se pueden solicitar desde la URL del explorador web para recibir un recurso. Y en la Figura 4.44 se muestra un pequeño ejemplo de un posible recuso disponible, en este caso se responde con una representación para un tipo de medio en formato HTML.

```

import org.restlet.Request;
import org.restlet.Response;
import org.restlet.Restlet;
import org.restlet.data.MediaType;
import org.restlet.representation.StringRepresentation;

/**
 *
 * @author Francisco José Benítez Pérez
 * Recurso Raíz.
 * Devuelve información sobre como usar WebHistoryFlow.
 */
public class WebHfRoot extends Restlet {

    @Override
    public void handle(Request request, Response response) {
        StringBuilder flujo = new StringBuilder();

        flujo.append("<html>");
        flujo.append("<head><title>WebHistoryFlow</title></head>");
        flujo.append("<body bgcolor=#E6E6E6>");
        flujo.append("</body>");
        flujo.append("</html>");

        response.setEntity(new StringRepresentation(
            flujo.toString(), MediaType.TEXT_HTML));
    }
}

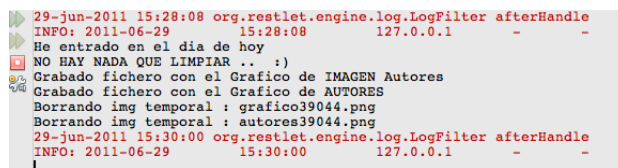
```

Figura 4.44: Recurso que devuelve un representación para un tipo de medio HTML.

### 4.3.3. Características comunes para recursos History Flow, Prefuse y gephi

Debido a la posibilidad de ejecución de varios hilos al mismo tiempo (conurrencia), se ha tenido que tener especial cuidado al almacenar el estado de las variables, con lo que se ha tenido que tener en cuenta los siguientes detalles :

- A la hora de generar cualquier tipo de fichero ya sea tipo gráficos o de datos (gml,xml), se ha ido creando un número aleatorio para adjuntarlo al nombre del gráfico o fichero generado, Figura 4.45, ya que la visualización de estos ficheros no tendrían porque realizarse inmediatamente después de su petición y por lo tanto podrían coincidir las peticiones en curso, con peticiones de recursos de otros usuarios. Con esto lo que se consigue es desvincular el proceso de generación del gráfico o fichero de datos, del proceso de visualización en el explorador del usuario.



```

29-jun-2011 15:28:08 org.restlet.engine.log.LogFilter afterHandle
INFO: 2011-06-29 15:28:08 127.0.0.1 - -
He entrado en el día de hoy
NO HAY NADA QUE LIMPIAR .. :)
Grabado fichero con el Grafico de IMAGEN Autores
Grabado fichero con el Grafico de AUTORES
Borrando img temporal : grafico39044.png
Borrando img temporal : autores39044.png
29-jun-2011 15:30:00 org.restlet.engine.log.LogFilter afterHandle
INFO: 2011-06-29 15:30:00 127.0.0.1 - -

```

Figura 4.45: Captura de pantalla del servidor, creando y borrando ficheros temporales.

- Se ha tenido que crear una clase que se encargue de eliminar todos los ficheros que se hubiesen creados y para ellos se ha definido un tiempo mas

que razonable para la eliminación de los ficheros en función del momento en que se crearon. Podríamos decir que un recolector de basuras.

- Todos los recursos requieren la ejecución previa de uno de los dos plugins encargados de los *artículos MediaWiki* o de los *ficheros de una svn*.
- Dentro de la clase *HistoryPanel* se generan todos los datos para formar un fichero *.xml*, *.gml* o un *gráfico*.
- Dentro de la aplicación *WebHistoryFlow* se establecen explícitamente algunos controles de error a la hora de especificar una URI por parte del usuario como :
  - Autor no indicado cuando se elige la opción *individual*.
  - Elección de modo de representación gráfica no definida.
  - Elección de opciones no compatibles con las definidas.
- Los tipos de medios usados a través de Restlet para los distintos recursos han sido:
  - Formato PNG para representación de gráficos History Flow (MediaType.IMAGE\_PNG).
  - Formato TXT para representaciones de texto ( MediaType.TEXT\_PLAIN ), como la salida del fichero *gml* para Gephi.
  - Formato HTML para representaciones de páginas web (MediaType.TEXT\_HTML), como la información de uso disponible al acceder al raíz de la aplicación.
  - Formato PDF para representaciones de gráficos Gephi (MediaType.APPLICATION\_PDF).
  - Formato XML para representaciones de ficheros xml para Prefuse (MediaType.APPLICATION\_XML).

#### 4.3.4. Adaptación, implementación History Flow a WebHistyFlow

Bien a grandes rasgos se van de describir las principales características de la adaptación de History Flow para escritorio a WebHistoryFlow para uso online.

Una vez analizado como funcionaba toda la aplicación en modo escritorio se pasa a eliminar todo interfaz de usuario ya que no sería necesario para la versión online. Una vez eliminadas las partes no necesarias se pasa a la adaptación de lo que se va a usar realmente definiendo para ello un método llamada

*CaminoWebHistoryflow()* en la clase *WebHistoryFlowMain*. *CaminoWebHistoryflow()* es el método encargado de instanciar los plugins de *History Flow* y/o *Richard Feard* que serán los que extraigan los datos de un artículo wiki o un fichero de una svn.

Dentro de la clase *WebHfServerApplication*, que gestiona la elección de los distintos recursos disponibles para los clientes, se define un método, *createInboundRoot()*, que es el encargado de mapear las clases a ejecutar en función del recurso solicitado por el cliente. Las opciones *History Flow* que se han implementado para su elección por parte de usuario, son las siguientes:

- Generar gráfico con formato *History Flow* de un artículo MediaWiki o de un fichero de una svn, sin la opción de un autor concreto de la última versión.
- Generar gráfico con formato *History Flow* de un artículo MediaWiki o de un fichero de una svn, indicando *modo individual* para referirse a un autor concreto de la última versión.

Cada uno de estos dos recursos incluye la posibilidad de indicar:

- Artículo o fichero svn a procesar.
- Autor para la opción individual.
- Modo de visualización (grupo, tiempo, link, reciente, individual).

Uno de los aspectos característicos de la aplicación original *History Flow* es que generaba en modo individual un gráfico con la representación de los Autores y otro gráfico con la representación del Flujo de Información de los distintos autores a lo largo de las versiones. Este tipo de enfoque no resultaba viable a la hora de mostrar un resultado en el explorador del usuario ya que le devolvería toda la información en dos bloques, rompiendo de esta forma la obtención de un solo recurso a partir de una URI's. Para ello se crea una clase llamada *UneImagenes* y desde su método *UneLasImgs()* recibe un vector con las imágenes a procesar.

A grandes rasgos el método *UneLasImgs()* lo que hace es definir un Buffer para imágenes, clase *BufferedImage* de Java, que tiene el tamaño de las dos imágenes a unir, a continuación se añade la primera imagen al principio del buffer y a continuación de esta la segunda, obteniendo posteriormente una nueva con la unión de ambas. La razón de realizarlo de esta manera es porque las clases que generaban dichos gráficos también realizaban al mismo tiempo otras operaciones y no resultaba práctico su modificación.

La implementación de los recursos para las opciones *History Flow* solo recogen aspectos de visualización de gráficos que son generados desde la clase

*HistoryPanel* calculado desde su método *doRenderBackground()* y gestionado finalmente por el método *paintComponent()* a través de un flujo de imagen el cual pasa en última instancia a un *tipo de medio Restlet*, *MediaType.IMAGE\_PNG*, para su representación.

#### 4.3.5. Implementar Prefuse y Gephi en WebHistoryflow

La inclusión de Prefuse y Gephi dentro de WebHistoryFlow ha sido con la idea de mostrar como se podían adaptar la información, ya sean en modo gráfico o en modo de datos, a una aplicación complementaria a WebHistoryFlow. Las opciones Prefuse y Gephi que se han implementado para su elección por parte de usuario, son las siguientes:

- Generar fichero *.xml* / *.gml* con formato Prefuse o Gephi, en función de un artículo MediaWiki.
- Generar fichero *.xml* / *.gml* con formato Prefuse o Gephi, en función de un fichero de una svn.
- A partir de una demo de Prefuse, mostrar gráfico tipo *applet* compuesto por nodos y aristas, estableciendo la relación Autores-Palabras de la última versión de un artículo MediaWiki o de un fichero de una svn.
- A partir de una demo de Gephi, mostrar gráfico tipo *pdf* compuesto por nodos y aristas, estableciendo la relación Autores-Palabras de la última versión de un artículo MediaWiki o de un fichero de una svn.

Para la ejecución del applet de Prefuse, descrito en la tercera opción de los puntos anteriores, se ha tenido que definir un *conector de Cliente CLAP*<sup>4</sup> de Restlet, necesario para permitir al cliente acceder a clases o ficheros de los recursos referidos a los applets a partir de un determinado directorio. Este acceso al directorio tal y como se ha implementado es el indicado para las peticiones GET que se realiza en la aplicación. Junto con esta definición se indica también a través de la implementación correspondiente la ruta para la ejecución del recurso.

Además para obtener una relación asociada a los datos que se procesaban dentro del gráfico applet de la demo de prefuse, se ha manipulado el código fuente incluyendo un nuevo color a los colores definidos de origen, con la idea de mostrar una información adicional, concretamente el color amarillo. Estos nuevos nodos se ha definido para :

- Creación de nuevos *nodos autores* en color amarillo con sus correspondientes aristas para representar relaciones *Autores - Palabras*.

---

<sup>4</sup>CLAP es un conector JVM para obtener la representación de los recursos accesibles a través de cargadores de clases y un esquema URI. CLAP es el acrónimo de : protocolo de acceso a cargador de clases.

- Creación de nuevos *nodos número* en color amarillo con sus correspondientes aristas para representar relaciones *Números - Palabras mas Repetidas*.

Para conseguir los ficheros necesarios para ser procesados por las correspondientes demos (*Prefuse, Gephi*), previamente se va generando, en función del artículo MediaWiki o del fichero de una svn, un fichero con formato xml para *Prefuse* y gml para *Gephi*. Estos ficheros se crean de manera dinámica a partir de artículo MediaWiki o del fichero de una svn y procesado por la clase *HistoryPanel*, que es la encargada de generar los datos. En el caso de tener que generar un gráfico para *Prefuse* o *Gephi* en vez de un fichero de datos, es éste fichero xml,gml el que se manda a la demo para la generación del gráfico y su posterior visualización en el explorador.

Desde la clase *HistoryPanel*, a través de los parámetros de su constructor se va a decidir que camino tomar a la hora de generar datos o gráficos, es decir :

- Gráfico formato History Flow.
- Fichero xml en formato Prefuse.
- Fichero gml en formato Gephi.

El método dentro la clase *HistoryPanel* encargado de realizar los cálculos es *doRenderBackground()*, desde este método se decide cual es la última versión del artículo mediawiki o del fichero de la svn elegido y una vez llegada a esta última versión se va almacenando en un objeto tipo *OutputStream*, donde se acumulan todas las líneas que se van creando hasta formar el fichero *xml* o *gml*.

Los ficheros de datos *xml,gml* que se generan en la clase *HistoryPanel* cuando se solicitan ficheros de datos de *Prefuse* o *Gephi*, están formados por unos identificadores de nodos y una relación entre ellos que serían las aristas.

Los nodos en el fichero generado *xml* de *Prefuse* están descritos por:

- Un identificador. Necesario para establecer relaciones entre nodos.
- Un atributo para definir el autor o la palabra no autor.
- Un atributo para definir el tipo y color del nodo, indicado a través de las letras *A, T, Y*. *A*, para indicar un *Autor*, *T*, para indicar una *Palabra que no sea Autor* y *Y*, para indicar un *nodo de color amarillo*.

Las aristas en el fichero generado *xml* de *Prefuse* están descritas por:

- Un identificador de nodo origen.



- Un identificador de nodo destino.

Los nodos en el fichero generado *gml* de Gephi están descritos por:

- Un identificador. Necesario para establecer relaciones posteriormente.
- Un atributo para definir el autor o la palabra no autor.

Las aristas en el fichero generado *xml* están descritas por:

- Un identificador de nodo origen.
- Un identificador de nodo destino.
- Un atributo valor. En este valor se ha indicado el numero de relaciones que se han establecido con esa misma relación.

#### 4.3.6. Interpretación de la representación gráfica de Prefuse y Gephi

Durante el proceso de análisis y diseño, una de las opciones que propuso el director del proyecto, fue la ampliación del proyecto base, que solo abarcaba gráficos con History Flow, con otros dos casos de prueba que emplease los datos que suministraba History Flow a través de su plugin o a través del plugin de Richard Feard para subversiones. Una vez seleccionados los frameworks a usar (Prefuse y Gephi), llegó el momento de buscar la relación a utilizar para darle sentido a los datos importados desde los artículos wiki o desde el fichero de una svn.

##### 4.3.6.1. Interpretación gráfico Prefuse

Teniendo en cuenta los *nodos* y las *aristas* que se muestran en la Figura 4.46, describiré la interpretación del gráfico tipo applet de la demo usada con el framework de *Prefuse*.

Para darle un mayor sentido a los nodos y aristas que mostraba el applet, se añadieron algunas líneas a la demo de prefuse, principalmente para que manipulase nodos con el color amarillo, los cuales representaría máximos en una determinada relación. Para intentar que no se mostrasen muchas palabras tipo determinante o artículo como (la, el , de etc..), solo se han tenido en cuenta aquellas palabras (tokens) cuyo tamaño era mayor que dos.

Se implementó el código necesario para crear, en función de los datos importados del artículo wiki o del fichero de la svn, un fichero de datos en formato XML para que lo interpretase la demo de Prefuse. En el fichero XML se creaban líneas de código donde se le iba dando a cada nodo un valor "Y", si correspondía, para decir que lo pintase en color amarillo, pues previamente se había

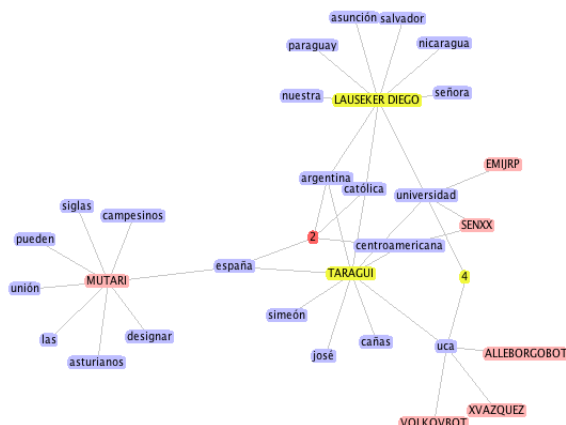


Figura 4.46: Representación gráfico applet demo Prefuse.

calculado que era un nodo con mas asociaciones que los demás en relación a los tipos de asociaciones establecidas en las líneas que indico a continuación.

Los *nodos* representados por *números* que indico en el gráfico Figura 4.46, aparecerán asociados solo a palabras y cada uno de ellos tiene el siguiente significado:

■ **Nodo -> Número.**

- Cuantos autores han mantenido la palabra que esta enlazada con el *nodo numero*, a lo largo de las distintas versiones-aportaciones, hasta llegar a la ultima version-aportación que es en la que estamos en este gráfico.
- Pintado en color amarillo lo(s) *nodos-número* que sean máximos respecto del punto anterior.

■ **Arista -> relación.**

Estas aristas establecen las relaciones entre nodos-numero con nodos-palabra interpretadas como:

- Quienes son las *palabras* con el *mismo numero* de autores que la han usado.

Los *nodos* representados por *autores* que indico en el gráfico Figura 4.46, aparecerán escritos en mayúsculas y asociados solo a palabras. Cada uno de ellos tiene el siguiente significado:

- **Nodo -> Autor.**

- Autor perteneciente a la última versión o aportación del artículo wiki o fichero de una svn.
- Pintado en color amarillo lo(s) *nodos-autor* que sean máximos respecto del punto anterior.

- **Arista -> relación.**

Estas aristas establecen las relaciones entre nodos-autor con nodos-palabra interpretadas como:

- Quienes son las *palabras* que están o han sido escritas por uno o mas autores.

**¿ Cómo se selecciona la palabra para asociarla como propia de un autor ?**

Teniendo en cuenta, en qué aportación-versión, de todas las aportaciones-versiones de esa wiki o fichero de svn, aparece por primera vez la palabra en cuestión y definimos que el primero que escribe una palabra, es el dueño de la misma.

A continuación describo alguna de las distintas relaciones representadas en el gráfico Figura 4.46.

- **Nodo : 2**

- **Palabra :** *españa*
- **Autores por los que ha pasado(mantenido) esa palabra hasta llegar a esta última aportación :** MUTARI , TARAGUI

- **Nodo : 2**

- **Palabra :** *católica*
- **Autores por los que ha pasado(mantenido) esa palabra hasta llegar a esta última aportación :** LAUSEKER DIEGO , TARAGUI
- **Palabras con el mismo numero de autores que la usan :** *españa , católica , centroamericana*

- **Nodo : 2**

- **Palabra :** *centroamericana*
- **Autores por los que ha pasado(mantenido) esa palabra hasta llegar a esta última aportación :** SENXX , TARAGUI
- **Palabras con el mismo numero de autores que la usan :** *españa , católica , centroamericana*

- **Nodo : 4**
  - **Palabra :** uca
  - **Autores por los que ha pasado(mantenido) esa palabra hasta llegar a esta última aportación :** TARAGUI , ALLEBOR-GOBOT , XVAZQUEZ , VOLKOVBOT
  - **Palabras con el mismo numero de autores que la usan :** uca , universidad
- **Nodo : 4**
  - **Palabra :** universidad
  - **Autores por los que ha pasado(mantenido) esa palabra hasta llegar a esta última aportación :** TARAGUI , LAUSEKER DIEGO , EMIJRP , SENXX
  - **Palabras con el mismo numero de autores que la usan :** uca , universidad

En este caso estos datos están sacados en función de la última aportación, de *XVAZQUEZ* con el siguiente texto :

- Uca es un género de crustáceos decápodos conocidos como cangrejos violinistas. Las siglas UCA pueden designar:
  - a la Universidad Católica "Nuestra Señora de la Asunción", en Paraguay
  - a la Universidad de Cádiz, en España;
  - a la Universidad Católica Argentina, en Argentina.
  - a la Universidad Centroamericana "José Simeón Cañas", en El Salvador.
  - a la Unión de Campesinos Asturianos, en España
  - de:UCA en:UCA fr:UCA it:UCA

#### 4.3.6.2. Interpretación gráfico Gephi

La representación que se devuelve al solicitar el gráfico de Gephi es similar a la anteriormente expuesta aunque con menos posibles interpretaciones y con distinto formato (*PDF*), como podemos ver en la Figura 4.47 , las principales diferencias de representación se encuentran en :

- Los nodos incrementan su tamaño en función del número de palabras que han escrito en la última versión del artículo wiki o fichero de svn. Sería similar a la cantidad de aristas que salen de los distintos autores del gráfico de Prefuse.

- Conclusión, en el gráfico del recurso Gephi, cuanto mayor sea el valor asignado a una palabra o a un autor, mayor será su tamaño o la intensidad del color. En el caso de la Figura 4.47, podemos ver que por ejemplo que el autor *TARAGUI* y la palabra *uca*, son una de las que mayor tamaño han alcanzado y que los colores que tienen asignados los nodos y sus aristas han incrementado la intensidad del color respecto a los demás.

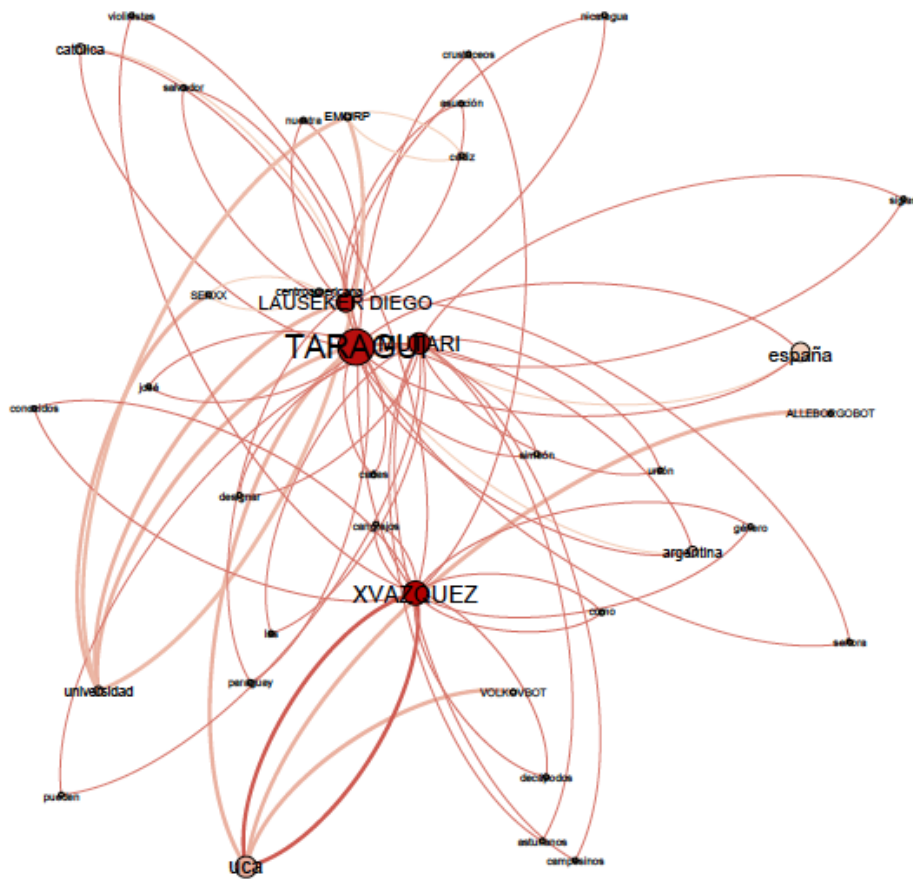


Figura 4.47: Representación gráfico pdf demo Gephi.



## Capítulo 5

# Evaluación y pruebas

En este capítulo se realizarán las pertinentes evaluaciones y pruebas en función de los tres módulos de que consta la aplicación WebHistoryFlow, es decir, formato History Flow, Gephi y Prefuse.

Se definirán unos pequeños conceptos fundamentales en la definición de casos de prueba, una descripción de la forma utilizada para su realización, los casos de pruebas analizados y un pequeño resumen de las pruebas realizadas.

### 5.1. Definiciones para los casos de prueba

Dentro del diccionario de IEEE <sup>1</sup> se recogen algunas definiciones en relación a las pruebas [13] a realizar durante y después de las implementaciones [IEEE, 1990], de entre las que relaciono las siguientes :

- **Prueba.** El nombre *prueba*, además de la actividad de probar, se puede usar para designar *un conjunto de casos y procedimientos de prueba* [IEEE, 1990].
- **Caso de prueba.** Un conjunto de entradas, condiciones de ejecución, y resultados esperados desarrollados para un objetivo particular como, por ejemplo, ejercitar un camino concreto de un programa o verificar el cumplimiento de un determinado requisito.
- **Defecto.** Un defecto en el software como, por ejemplo, un proceso, una definición de datos o un paso de procesamiento incorrectos en un programa.
- **Fallo.** La incapacidad de un sistema o de alguno de sus componentes para realizar las funciones requeridas dentro de los requisitos de rendimientos especificados.

---

<sup>1</sup>Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.

- **Error**, con varias acepciones:
  - La diferencia entre un valor calculado, observado o medido y el valor verdadero, especificado o teóricamente correcto.
  - Un defecto.
  - Un resultado incorrecto.
  - Una acción humana que conduce a un resultado incorrecto.

## 5.2. Diseños para los casos de prueba

A la hora de establecer el enfoque para el diseño de los casos de prueba, se utilizan unas determinadas técnicas cuyo objetivo consiste en dar una cierta confianza en que se detectarán los defectos existentes sin que se necesite consumir una cantidad excesiva de recursos. En la Figura 5.1 se puede observar el ciclo completo de las pruebas.

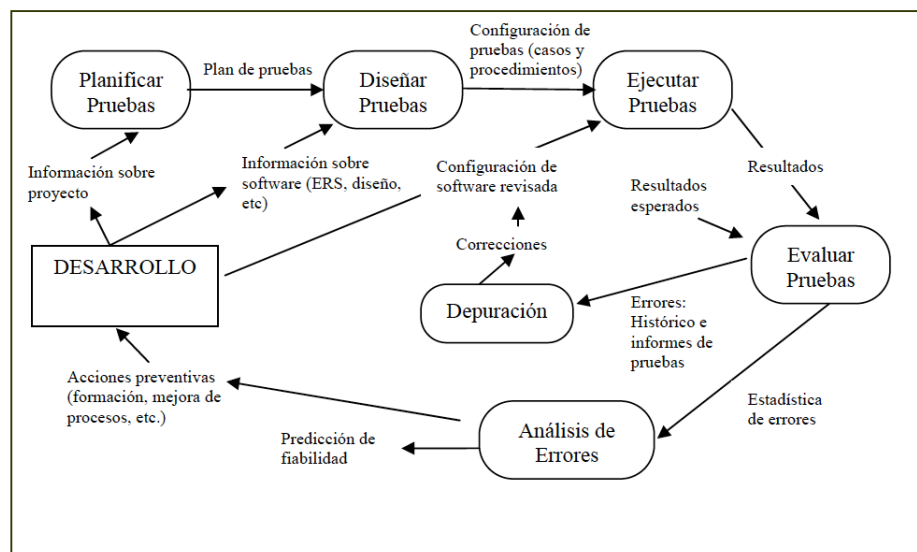


Figura 5.1: Ciclo completo de las pruebas [13].

Los tres enfoque principales existentes para el diseño de casos de prueba:

- *Enfoque estructural o de caja blanca*, que se centra en la estructura interna (implementación) del programa para elegir los casos de prueba.
- *Enfoque funcional o de caja negra*, que estudia la especificación de las funciones, la entrada y la salida para derivar los casos.



- Enfoque aleatorio utilizando un modelo, en muchas ocasiones estadísticos, para representar las posibles entradas al programa para crear a partir de ellos los casos de prueba.

Para este proyecto por sus características nos centraremos en realizar las principales casos de prueba en el aspecto de entradas de peticiones y salidas de recursos realizados por los usuarios, por lo tanto usaremos un enfoque funcional o de caja negra.

### 5.3. Casos de prueba

Para establecer una relación de los distintos casos de prueba he optado por ajustarme lo máximo posible al estándar de la IEEE Std 829, el cual establece criterios de representación a través de unos documentos tipo, donde se representa una relación de las distintas pruebas a realizar en un determinado sistema.

Algunos de los apartados que se especifican en los distintos documentos del estándar no serán descritos por no ser relevantes o haberse expuesto en mayor o menor grado en apartados anteriores de la memoria.

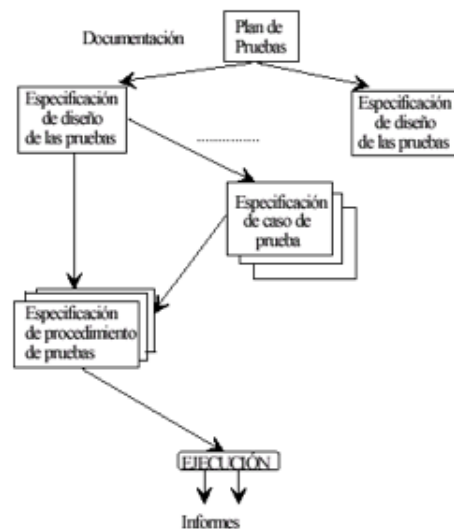


Figura 5.2: Ciclo pruebas estándar IEEE 829.

Todos los casos de prueba que se exponen se han realizado desde un entorno local, es decir, tomando como *host* el servidor local de mi equipo a través del puerto 9080. Asumo que las condiciones de ejecución en un servidor remoto serán similares a las expuestas salvo lo referido a los tiempos de visualización, debido a las posibles diferencias en el ancho de banda en mi red y en la de la UCA así como la diferencia en velocidad de ejecución de las aplicaciones

en el servidor Apache de mi equipo con respecto al equipo con la instalación Tomcat de Apache existente en la UCA.

Aplicaremos el estándar de la IEEE Std.829, como se puede ver en la Figura 5.2, para detallar los siguientes documentos:

- Plan de Pruebas
- Especificación del diseño de pruebas
- Especificación de caso de prueba
- Especificación de procedimiento de prueba

Cuando las pruebas a realizar están enfocadas a un desarrollo orientado a objetos, como es nuestro caso, las estrategias y técnicas a usar difieren a las de un enfoque estructurado. Las técnicas de caja negra se centrarán en el comportamiento de la aplicación. Las pruebas de sistema, serán las descripciones de los casos de uso, que van a ser las que aporten las referencias para la elección de los casos de prueba, partiendo de los datos y los eventos incluidos en los escenarios.

### 5.3.1. Diseño de los casos de prueba según el estándar IEEE Std. 829

Para la documentación de los casos de prueba se especifican los siguientes documentos con sus correspondientes apartados, a partir de ellos se podrán consultar las pruebas realizadas para obtener un producto que cumpla y satisfaga las necesidades del objetivo del proyecto.

Para dar constancia de las pruebas realizadas y de la finalización de las mismas positivamente, se incluye con cada caso de prueba, un gráfico del resultado obtenido.

#### 5.3.1.1. Plan de pruebas

##### 1. Identificador del plan de prueba.

*WEBHF*

##### 2. Introducción y resumen de elementos a probar.

- El documento a continuación identifica elementos de prueba, las características que deben probarse, las tareas de prueba, lo que hará cada tarea.
- El elemento a ser probado es el software.

##### 3. Elementos - Software que se va a probar.

MÓDULO DE ENTRADA

- Sub. Módulo - Información de uso de la aplicación:  
Se indica una URI con el root del servidor en el explorado web.
- Sub. Módulo - Gráficos History Flow:  
Se indica URI con parámetros para formato History Flow.
- Sub. Módulo - Gráficos o Datos Prefuse:  
Se indica URI con parámetros para formato Prefuse.
- Sub. Módulo - Gráficos o Datos Gephi:  
Se indica URI con parámetros para formato Gephi.

#### MÓDULO DE SALIDA

- Sub. Módulo - Visualización pagina web con información de uso de la aplicación.
- Sub. Módulo - Visualización gráfico con formato History Flow en el explorador web.
- Sub. Módulo - Visualización gráfico o datos con formato Prefuse en el explorador web.
- Sub. Módulo - Visualización gráfico o datos con formato Gephi en el explorador web.

#### 4. Características que se van a probar.

- fluidez de datos
- Independencia de módulos
- Soporte del software

#### 5. Características que no se prueban.

- Errores relacionados con el tiempo.
- Errores de datos no coherentes a la hora de importar el fichero XML.
- Condiciones de error no detectadas.
- Condiciones especiales de los datos de entrada en los ficheros XML de los artículos wiki o ficheros de una svn.
- Invalidez de la información mostrada por pantalla.
- Interacción con tareas en background.
- Fallos de configuración/compatibilidad con software.
- Incapacidad de soportar el volumen de carga o fallos hardware.

#### 6. Enfoque general de la prueba.

- **Prueba de Diseño.** Se asegura que la arquitectura propuesta sea coherente, consistente y completa
- **Prueba de Unidad.** Se prueban que los fragmentos individuales (unidades) que forman el sistema cumplen las especificaciones y tienen el comportamiento esperado.
- **Prueba de Requisitos.** Se validan métodos y procesos para comprobar requisitos (prototipos, casos de uso, etc.).
  - Comprobación de la compleción(completo) y consistencia.
  - Eliminación de requisitos duplicados.
- **Prueba de Integración.** Se prueban las funcionalidades, rendimiento, fiabilidad, etc. del sistema, sus relaciones con el exterior, etc.
- **Prueba de Sistema.** Se ha comprobado que el sistema software y hardware está integrado cumpliendo lo requisitos funcionales exigidos para el proyecto. También se ha comprobado la adecuación de la documentación de usuario y la ejecución en condiciones límite y de sobrecarga, ejecutando varios hilos simultáneamente para solicitar distintos recursos.

#### 7. Criterios de paso/fallo para cada elemento.

Los criterios de paso para cada módulo consistirá en que cada vez que un usuario indica una URI, si ésta coincide con la ruta de alguno de los recursos se visualizará en el explorador web. Los criterios de fallo serán aquellos donde por el contrario no coincide la ruta y el framework de Restlet indicará el error pertinente, así como algunos tipos de errores definidos implícitamente en función del recurso solicitado.

#### 8. Criterios de suspensión y requisitos de reanudación.

Las pruebas deberán ser suspendidas si no se tienen operativos los siguientes elementos software:

- Aplicación WebHistoryWeb corriendo en el servidor Tomcat de Apache.
- Explorador web.
- Máquina Virtual de Java operativa en el explorador, ver. 1.6
- Conexión a internet.

#### 9. Documentos a proporcionar.

- Plan de Pruebas.
- Especificación del diseño de pruebas.
- Especificación de casos de prueba.

- Especificación de procedimiento de pruebas.

10. **Actividades de preparación y ejecución de pruebas.**

FRANCISCO JOSÉ BENÍTEZ PÉREZ

- Preparación de casos de prueba.
- Ejecución de pruebas.
- Datos de la prueba.
- Preparar informe.

11. **Necesidades de entorno.**

En cuanto a *Software* y *Hardware* :

- Se requieren dos máquinas. Una será el servidor y otra será el cliente.
- Sistema operativo MacOSx, Linux o Windows.
- Explorador web.
- Conexión a internet.
- Máquina virtual de Java ver. 1.6

12. **Responsabilidades en la organización y realización de las pruebas.**

Pruebas de software: Francisco José Benítez Pérez.

13. **Necesidades de personal y de formación.**

Básicas, nivel usuario.

### 5.3.1.2. Especificación de diseño de pruebas

#### 1. Identificador diseño de pruebas.

*WEBHF-01*

#### 2. Características a probar de los elementos software.

- Fluidez de datos.
- Independencia de módulos.
- Soporte del software.

#### 3. Identificaciones de cada prueba.

- *WEBHF-01-dis-01*
  - Solicitar información de uso de la aplicación.
  - Se indicará en el explorador web una URI con el root del servidor.
- *WEBHF-01-dis-02*
  - Solicitar gráfico con formato HF(History Flow) de un artículo MediaWiki en modo grupo, link, reciente y tiempo.
  - Se indicará en el explorador web una URI indicando *"host\_local/wiki/host\_wiki/articulo/modo"*.
- *WEBHF-01-dis-03*
  - Solicitar gráfico con formato HF de un artículo MediaWiki para un autor concreto de la última versión de ese artículo wiki.
  - Solicitar gráfico con formato HF de un artículo MediaWiki para un autor concreto de la última versión de ese artículo wiki.
  - Solicitar gráfico con formato HF de un artículo MediaWiki para un autor concreto de la última versión de ese artículo wiki.
  - Se indicará en el explorador web una URI indicando *"host\_local/wiki/host\_wiki/articulo/individual/autor"*.
- *WEBHF-01-dis-04*
  - Solicitar gráfico con formato HF de un fichero de una subversión en modo grupo, link, reciente y tiempo.
  - Se indicará en el explorador web una URI indicando *"host\_server/svn/host\_svn/camino/fichero/modo"*.
- *WEBHF-01-dis-05*
  - Solicitar gráfico con formato HF de un fichero de una subversión para un autor concreto de la última versión de ese fichero.
  - Se indicará en el explorador web una URI indicando *"host\_server/svn/host\_svn/camino/fichero/individual/autor"*.

- *WEBHF-01-dis-06*
  - Solicitar gráfico con formato Gephi de un artículo MediaWiki.
  - Se indicará en el explorador web una URI indicando *"host\_local/wiki/host\_wiki/articulo/modo"*.
- *WEBHF-01-dis-07*
  - Solicitar gráfico con formato Gephi de un fichero de una subversión.
  - Se indicará en el explorador web una URI indicando *"host\_server/svn/gephi/host\_svn/camino/fichero"*.
- *WEBHF-01-dis-08*
  - Solicitar fichero de datos con formato Gephi de un artículo MediaWiki.
  - Se indicará en el explorador web una URI indicando *"host\_server/wiki/gephi/host\_wiki/articulo/gml"*.
- *WEBHF-01-dis-09*
  - Solicitar fichero de datos con formato Gephi de un fichero de una subversión.
  - Se indicará en el explorador web una URI indicando *"host\_server/svn/gephi/host\_svn/camino/fichero/gml"*.
- *WEBHF-01-dis-10*
  - Solicitar gráfico con formato Prefuse de un artículo MediaWiki.
  - Se indicará en el explorador web una URI indicando *"host\_server/wiki/prefuse/host\_wiki/articulo/"*.
- *WEBHF-01-dis-11*
  - Solicitar gráfico con formato Prefuse de un fichero de una subversión.
  - Se indicará en el explorador web una URI indicando *"host\_server/svn/prefuse/host\_svn/camino/fichero/"*.
- *WEBHF-01-dis-12*
  - Solicitar fichero de datos con formato Prefuse de un artículo MediaWiki.
  - Se indicará en el explorador web una URI indicando *"host\_server/wiki/prefuse/host\_wiki/articulo/xml"*.
- *WEBHF-01-dis-13*
  - Solicitar fichero de datos con formato Prefuse de un fichero de una subversión.
  - Se indicará en el explorador web una URI indicando *"host\_server/svn/prefuse/host\_svn/camino/fichero/xml"*.

- *WEBHF-01-dis-14*
    - Solicitar gráfico con formato HF de un artículo MediaWiki o un fichero de una svn indicando el modo individual sin incluir autor.
    - Se indicará en el explorador web una URI indicando *"host\_local/wiki/host\_wiki/host\_svn/[camino]/articulo/fichero/individual"*.
  - *WEBHF-01-dis-15*
    - Solicitar gráfico con formato HF de un artículo MediaWiki o un fichero de una svn indicando parámetros no coincidentes con rutas definidas para la obtención de recursos.
    - Se indicará en el explorador web una URI indicando *"host\_local/noexiste/host\_wiki/host\_svn/[camino]/articulo/fichero/modo"*.
4. **Criterios de paso/fallo de la prueba(criterios para determinar si una característica o combinación de características ha pasado con éxito la prueba o no).**
- La regla general para determinar si la prueba ha pasado o no con éxito, será la visualización de los datos o gráficos en el explorador o la visualización de un gráfico vacío de color negro o la ausencia de datos transcurrido un tiempo no mayor a 10 segundos en condiciones normales.
- La única excepción sería para la visualización de gráficos en formato Pre-fuse en el que el tiempo de visualización iría en proporción al volumen de datos a procesar. Para un fichero con pocos datos estaría entre 15 y 20 segundo para su visualización.



### 5.3.1.3. Especificación de casos de prueba

#### 1. Identificador.

WEBHF-01-dis-01

#### 2. Elementos software que se van a probar.

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfRoot.class
- Visualización correcta de la información en formato HTML.

#### 3. Especificación de la entrada.

Se indica en el explorador web la siguiente URI:

*http://localhost:9080*

#### 4. Especificación de la salida.

Visualización de página web con las instrucciones de uso como se muestra en la Figura 5.3.

Proyecto Fin de Carrera :: Servicio web para la visualización gráfica de las contribuciones en webs colaborativas					
Francisco José Benítez Pérez					
UCA - PFC - 2011 WebHistoryFlow	Recurso Wiki		Tabla Guía para la utilización de WebHistoryFlow		
				Recurso Svr	
Gráfico HistoryFlow	host_server/wiki/host_wiki/articulo/modo	<a href="http://wiki.wikipedia.org/CA/ajuste">http://wiki.wikipedia.org/CA/ajuste</a>	host_server/vn/host_servicios/fichero/modo	<a href="http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste">http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste</a>	
Gráfico HistoryFlow con Autor	host_server/wiki/host_wiki/articulo/individual/autor	<a href="http://wiki.wikipedia.org/CA/individual/ajuste">http://wiki.wikipedia.org/CA/individual/ajuste</a>	host_server/vn/host_servicios/fichero/individual/autor	<a href="http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/individual/ajuste">http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/individual/ajuste</a>	
Gráfico Graph	host_server/wiki/graph/host_wiki/articulo	<a href="http://wiki.wikipedia.org/CA/">http://wiki.wikipedia.org/CA/</a>	host_server/vn/graph/host_servicios/fichero	<a href="http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/graph">http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/graph</a>	
Salida fichero XML - Graph	host_server/wiki/graph/host_wiki/articulo/xml	<a href="http://wiki.wikipedia.org/CA/xml">http://wiki.wikipedia.org/CA/xml</a>	host_server/vn/graph/host_servicios/fichero/xml	<a href="http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/graph/xml">http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/graph/xml</a>	
Gráfico Aplicar nodos-artistas Prefuse	host_server/wiki/prefuse/host_wiki/articulo	<a href="http://wiki.wikipedia.org/CA/">http://wiki.wikipedia.org/CA/</a>	host_server/vn/prefuse/host_servicios/fichero	<a href="http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/prefuse">http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/prefuse</a>	
Salida fichero XML - Prefuse	host_server/wiki/prefuse/host_wiki/articulo/xml	<a href="http://wiki.wikipedia.org/CA/xml">http://wiki.wikipedia.org/CA/xml</a>	host_server/vn/prefuse/host_servicios/fichero/xml	<a href="http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/prefuse/xml">http://webhistory.asamblea.com/vn/ajuste/Akactivismo/Akactivismo/Modelo/ajuste/prefuse/xml</a>	

Significado de las palabras en negrita :

- hostwiki** : Host de artículos tipo wikipedia, como :
  - en.wikipedia.org
  - meta.wikipedia.org
  - www.mediawiki.org
- hostserv** : Host de las svr que contiene el fichero, como :
  - www.quepasa.org
- articulo** : Nombre del artículo wiki.
- fichero** : Nombre del fichero de una svr que queremos analizar.
- modo** : Orden de modo de visualización del gráfico webhistoryflow, hay predefinidos cinco modos de visualización.
  - grupo (Representa las contribuciones de todos los autores a través de colores).
  - tiempo (Representa la persistencia de las contribuciones a través del tiempo).
  - link (Muestra la colocación de los links sobre las páginas y su persistencia a lo largo del tiempo).
  - reciente (Resalta la página correspondiente a los ficheros actualizados recientemente en cada versión).
  - individual (Resalta la contribución de un autor a través de todas las versiones).
- autor** : Nombre de uno de los autores que pertenecen a una wiki o participan en un fichero de una svr.

Opciones generales :

- Gráfico HistoryFlow** : Visualiza gráficos en formato HistoryFlow, representando el flujo de vida de una wiki o una svr, así como los autores que participan en dicha wiki o svr, clasificado por colores.
- Gráfico HistoryFlow con Autor** : Idem a lo anterior pero con la posibilidad de indicar un autor concreto de los participantes en una wiki o fichero de la svr, resaltando su aportación durante todas las versiones.
- Gráfico Graph** : Devuelve un gráfico en formato PDF a partir de una implementación demo descargable desde la web de Graph, pero a través de los datos proporcionados por WebHistoryFlow.
- Salida fichero XML - Graph** : Devuelve un fichero en formato xml que es usado por la API de Graph, a partir de una fichero pasado completo en la aplicación de exportación generada en su web y realizar el tipo de gráfico que más te guste.
- Gráfico Aplicar nodos-artistas Prefuse** : Visualiza un apilado en el cual se visualiza una representación gráfica a partir de nodos y aristas, de la última versión de una wiki o una svr, mostrando la relación entre los autores de esa versión y las palabras escritas por ellos, así como el autor(es) que más palabras proporcionan y las palabras que más se repiten, entre dos últimas características se representan con nodos de color amarillo. Así mismo al posicionarse el ratón sobre cualquier nodo se activan los que están relacionadas.
- Salida fichero XML - Prefuse** : Devuelve un fichero en formato xml en el cual se puede observar los distintos modos que representan dicha versión de wiki o svr, así como las aristas que los unen. Igualmente puede ser usado para usar en alguna implementación gráfica de la API Prefuse.

Figura 5.3: Recurso : Página Web información uso aplicación WebHistoryFlow.

### 5. Necesidades de entorno.

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.

### 6. Requisitos especiales de procedimiento.

Ninguno.

### 7. Dependencias entre casos de prueba.

Este caso de prueba no depende de otro caso.

**1. Identificador.**

*WEBHF-01-dis-02*

**2. Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfWikiServerRecurso.class
- Ejecución del recurso en modo : *grupo*, *link*, *tiempo* y *reciente*.
- Unión correcta de las imágenes *autor* y *flujo* en una única.
- Salida correcta de un solo gráfico, en formato PNG.
- Flujo de información servidor-cliente aceptable.

**3. Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*<http://localhost:9080/wiki/es.wikipedia.org/UCA/grupo>*

**4. Especificación de la salida.**

Visualización del gráfico con formato HF, de un artículo Mediawiki ,representando un flujo de información a través de un gráfico de paralelogramos y los autores que intervienen, en cuatro de sus cinco modos de visualización :

- *grupo* - Figura 5.4.
- *link* - Figura 5.5.
- *tiempo* - Figura 5.6.
- *reciente* - Figura 5.7.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización artículo wiki.

**6. Requisitos especiales de procedimiento.**

Ninguno.

**7. Dependencias entre casos de prueba.**

Este caso de prueba no depende de otro caso.

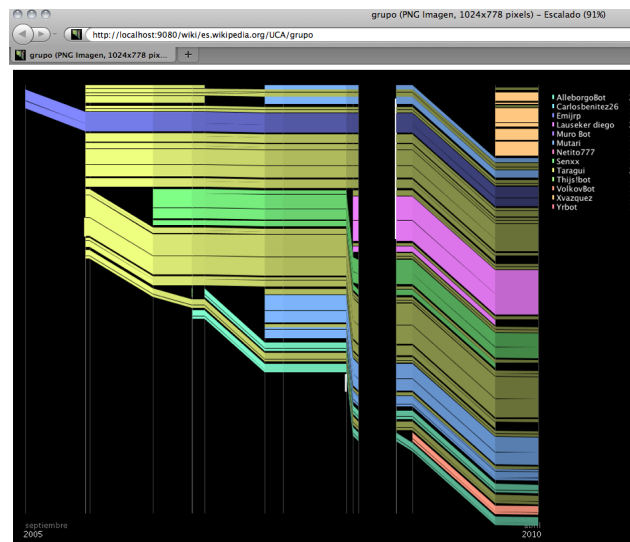


Figura 5.4: Recurso : Gráfico artículo wiki con formato History Flow, modo: *grupo*.

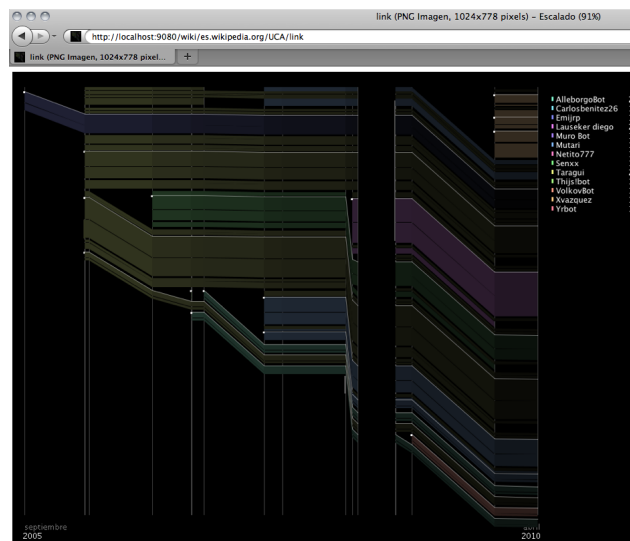


Figura 5.5: Recurso : Gráfico artículo wiki con formato History Flow, modo: *link*.

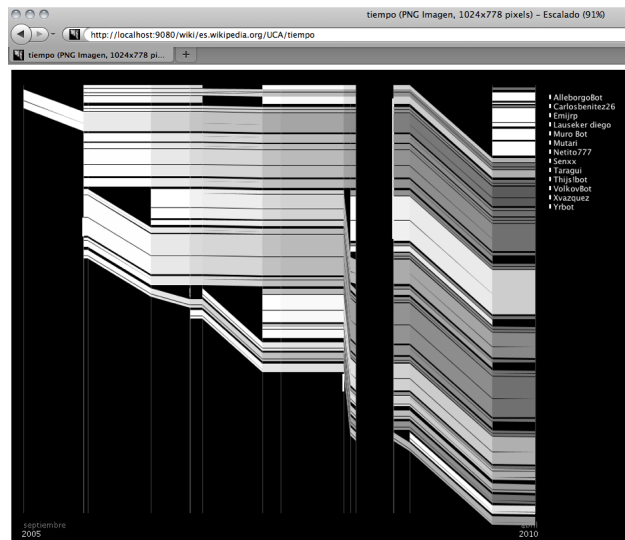


Figura 5.6: Recurso : Gráfico artículo wiki con formato History Flow, modo: *tiempo*.

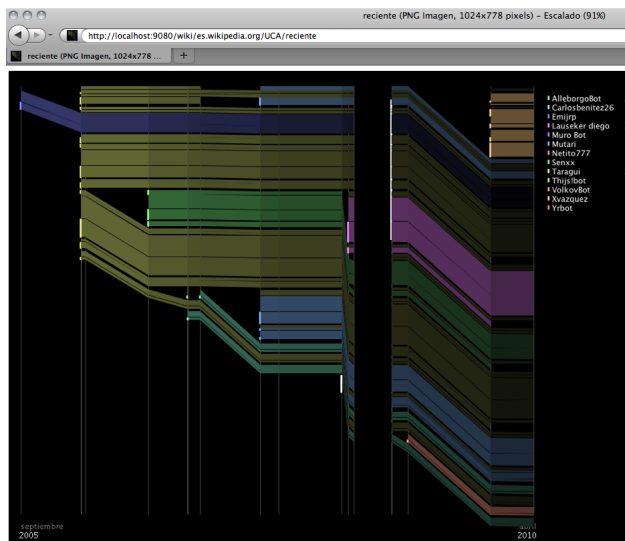


Figura 5.7: Recurso : Gráfico artículo wiki con formato History Flow, modo: *reciente*.

1. **Identificador.**

*WEBHF-01-dis-03*

2. **Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfWikiAutorServerRecurso.class
- Ejecución del recurso en modo : *individual*.
- Unión correcta de las imágenes *autor* y *flujo* en una única.
- Salida correcta de un solo gráfico, en formato PNG.
- Flujo de información servidor-cliente aceptable.
- Resalta correctamente el autor seleccionado en el flujo de información.
- Resalta correctamente el autor elegido de entre los existentes.

3. **Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

<http://localhost:9080/wiki/es.wikipedia.org/UCA/individual/Emijrp>

4. **Especificación de la salida.**

Visualización del gráfico con formato HF, de un artículo Mediawiki referido a uno de los autores ,representando un flujo de información a través de un gráfico de paralelogramos y un gráfico con los autores que intervienen, en su modo de visualización *individual* como se muestra en la Figura 5.8.

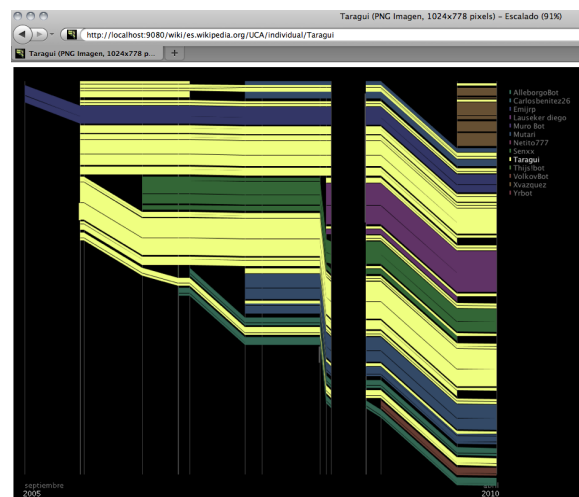


Figura 5.8: Recurso : Gráfico artículo wiki con formato History Flow de un autor, modo: *individual*.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización artículo wiki.

**6. Requisitos especiales de procedimiento.**

Ninguno.

**7. Dependencias entre casos de prueba.**

Este caso de prueba no depende de otro caso.

**1. Identificador.**

*WEBHF-01-dis-04*

**2. Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfSvnServerRecurso.class
- Ejecución del recurso en modo : *grupo*, *link*, *tiempo* y *reciente*.
- Unión correcta de las imágenes *autor* y *flujo* en una única.
- Salida correcta de un solo gráfico, en formato PNG.
- Flujo de información servidor-cliente aceptable.
- Ejecución correcta del parámetro *camino* que se especifica separado por puntos.
- Ejecución correcta en distintos servidores de subversiones.

**3. Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*http://localhost:9080/svn/subversion.assembla.com/  
svn.iff6.trunk.Akavriumas/Akvariumas Model.emx/grupo*

**4. Especificación de la salida.**

Visualización del gráfico con formato HF, de un fichero de una svn ,representando un flujo de información a través de un gráfico de paralelogramos y los autores que intervienen, en cuatro de sus cinco modos de visualización :

- *grupo* - Figura 5.9.
- *link* - Figura 5.10.
- *tiempo* - Figura 5.11.
- *reciente* - Figura 5.12.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización de un fichero de svn.

### 6. Requisitos especiales de procedimiento.

Para evitar problemas a la hora de proporcionar a Restlet los parámetros adecuados para establecer la ruta del recurso correspondiente he optado por dividir el camino completo al fichero y desglosarlo en dos partes. Una sería el dominio del servidor y la otra el camino hasta el fichero elegido. Y como requisito para ello es obligatorio especificar el parámetro *camino*, separado por puntos. Ejem.: *svn.iff6.trunk.Akavriumas*

### 7. Dependencias entre casos de prueba.

Este caso de prueba no depende de otro caso.

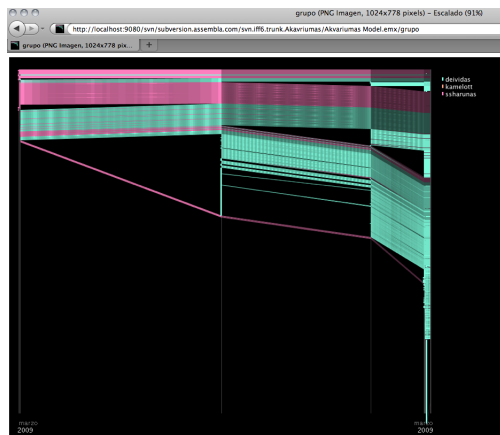


Figura 5.9: Recurso : Gráfico fichero svn con formato History Flow, modo: *grupo*.

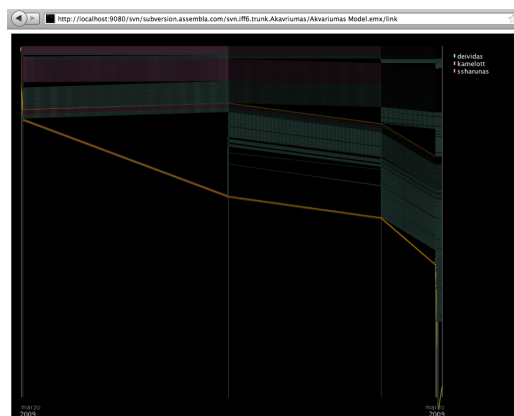


Figura 5.10: Recurso : Gráfico fichero svn con formato History Flow, modo: *link*.



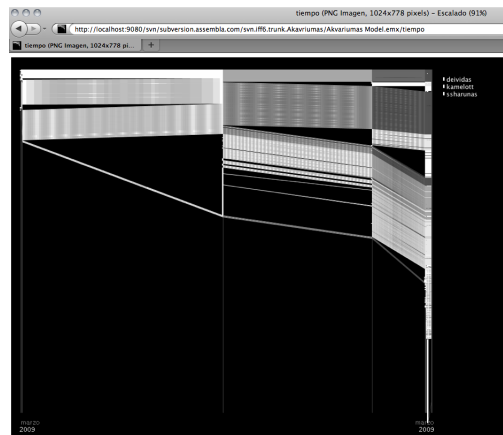


Figura 5.11: Recurso : Gráfico fichero svn con formato History Flow, modo: *tiempo*.

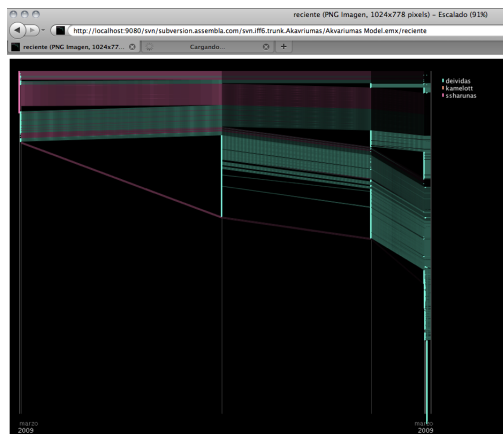


Figura 5.12: Recurso : Gráfico fichero svn con formato History Flow, modo: *reciente*.

**1. Identificador.**

*WEBHF-01-dis-05*

**2. Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfSvnAutorServerRecurso.class
- Ejecución del recurso en modo : *individual*.
- Unión correcta de las imágenes *autor* y *flujo* en una única.
- Salida correcta de un solo gráfico, en formato PNG.
- Flujo de información servidor-cliente aceptable.
- Resalta correctamente el autor seleccionado en el flujo de información.
- Resalta correctamente el autor elegido de entre los existentes.

**3. Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*http://localhost:9080/svn/subversion.assembla.com/*

*svn.iff6.trunk.Akavriumas/Akvariumas Model.emx/individual/devidas*

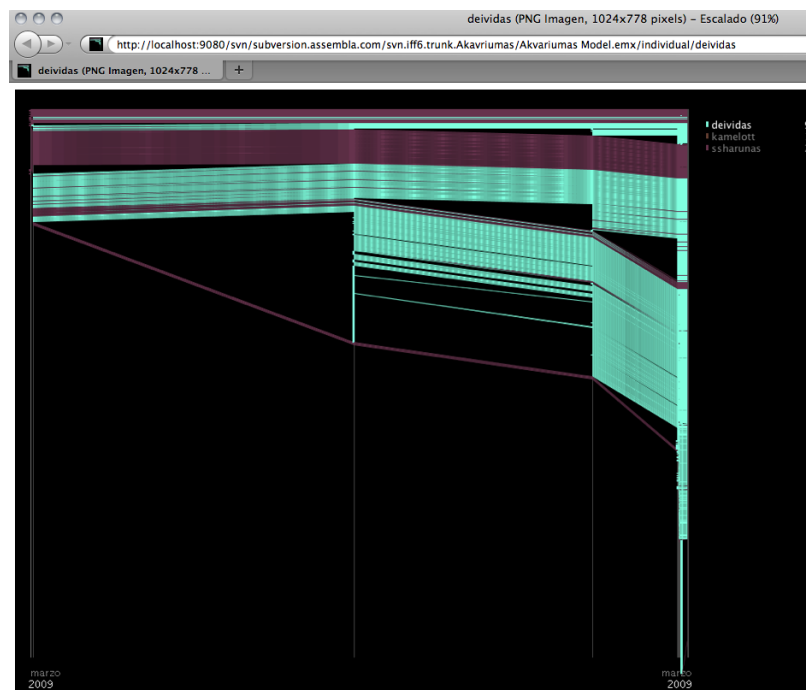


Figura 5.13: Recurso : Gráfico formato History Flow de un de los autores del fichero perteneciente a una svn, modo: *individual*.

**4. Especificación de la salida.**

Visualización del gráfico con formato HF, de un fichero de una svn referido a uno de los autores ,representando un flujo de información a través de un gráfico de paralelogramos y un gráfico con los autores que intervienen, en su modo de visualización *individual* como se muestra en la Figura 5.13.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización de un fichero de svn, Figura 5.14.

**6. Requisitos especiales de procedimiento.**

Indicar el parámetro *camino* separado por puntos.

**7. Dependencias entre casos de prueba.**

Este caso de prueba no depende de otro caso.

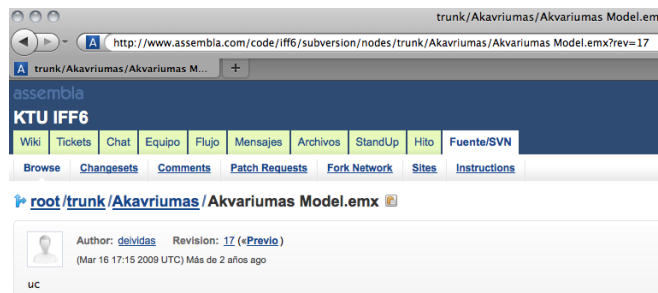


Figura 5.14: Muestra de directorio de un fichero en *www.assembla.com*

**1. Identificador.**

*WEBHF-01-dis-06*

**2. Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfWikiGephiServerRecurso.class
- Visualización correcta de la representación de nodos y aristas en formato *pdf*.
- Flujo de información servidor-cliente aceptable.
- Generación en formato correcto del fichero temporal de datos con formato *gml* para su posterior uso en la demo de Gephi.
- Comprobación de coherencia de los datos de origen con respecto a los gestionados.
- Similitud en cuanto al resultado del gráfico con respecto al gráfico en formato HF.

**3. Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*<http://localhost:9080/wiki/gephi/es.wikipedia.org/UCA>*

**4. Especificación de la salida.**

Visualización del gráfico con formato demo Gephi, de un artículo Mediawiki ,representando un flujo de información a través de nodos y aristas en función de la última versión del artículo mediawiki como puede verse en la Figura 5.15.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización artículo wiki.

## 6. Requisitos especiales de procedimiento.

Ninguno.

## 7. Dependencias entre casos de prueba.

Este caso de prueba no depende de otro caso.

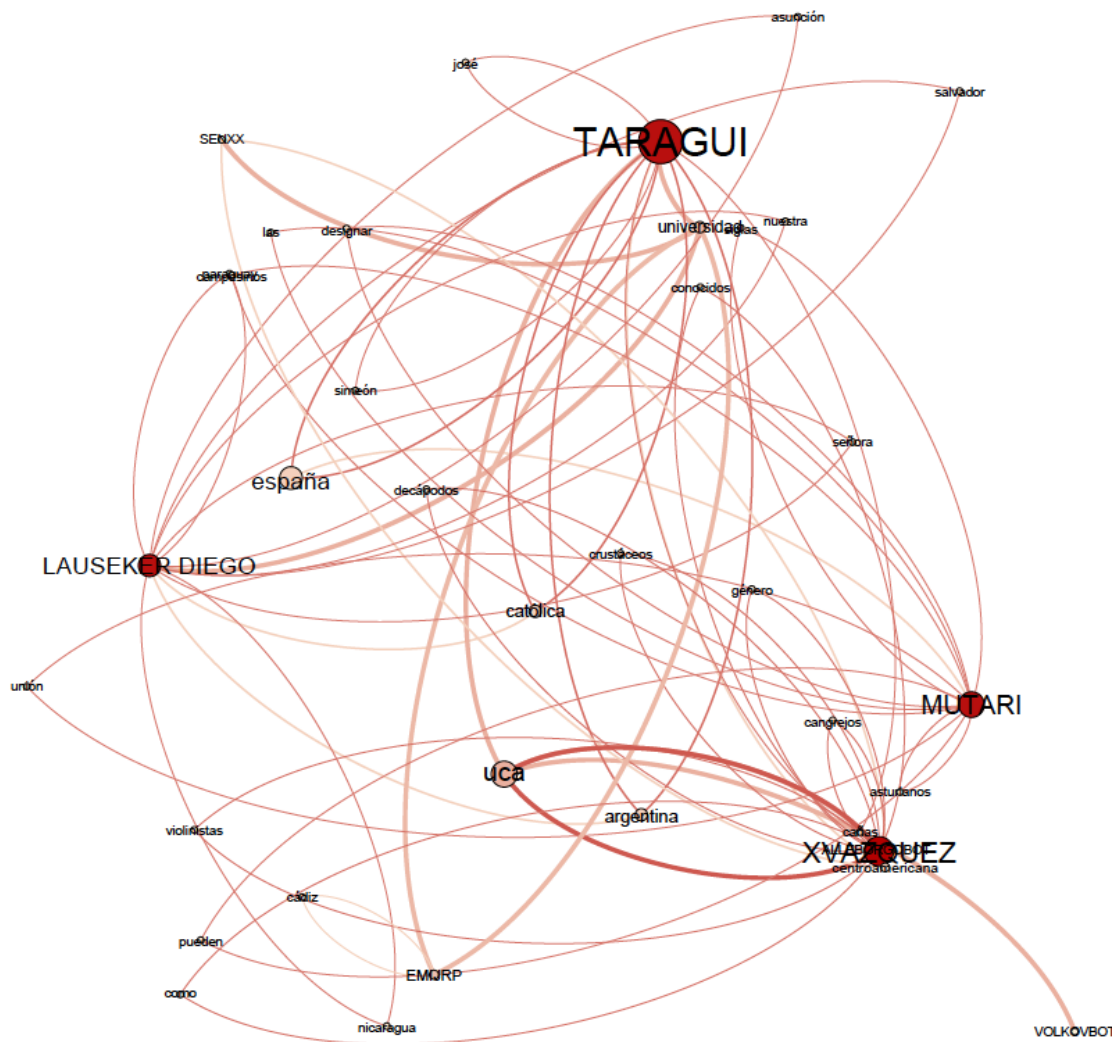


Figura 5.15: Recurso : Gráfico artículo wiki con formato Gephi.

**1. Identificador.**

*WEBHF-01-dis-07*

**2. Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfSvnGephiServerRecurso.class
- Visualización correcta de la representación de nodos y aristas en formato *pdf*.
- Flujo de información servidor-cliente aceptable.
- Generación en formato correcto del fichero temporal de datos con formato *gml* para su posterior uso en la demo de Gephi.
- Comprobación de coherencia de los datos de origen con respecto a los gestionados.
- Similitud en cuanto al resultado del gráfico con respecto al gráfico en formato HF.
- Flujo de información servidor-cliente aceptable.
- Ejecución correcta del parámetro *camino* que se especifica separado por puntos.
- Ejecución correcta en distintos servidores de subversiones.

**3. Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*http://localhost:9080/svn/gephi/svn.apache.org/  
repos.asf.subversion.trunk.notes/xfail-status*

**4. Especificación de la salida.**

Visualización del gráfico con formato demo Gephi, de un fichero perteneciente a una svn ,representando un flujo de información a través de nodos y aristas en función de la última versión del fichero como puede verse en la Figura 5.16.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.

- Aplicación WebHistoryFlow en ejecución.
  - Localización de un fichero de svn.
6. **Requisitos especiales de procedimiento.**  
Indicar el parámetro *camino* separado por puntos.
7. **Dependencias entre casos de prueba.**  
Este caso de prueba no depende de otro caso.

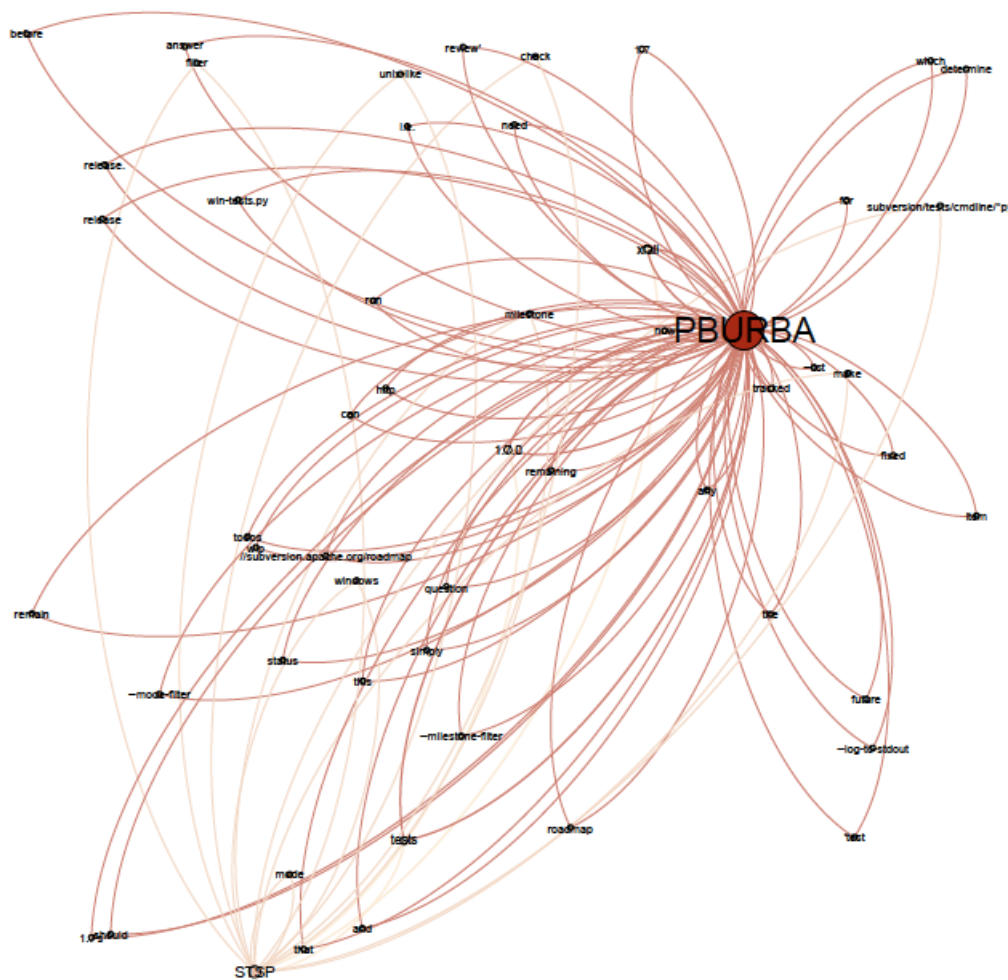


Figura 5.16: Recurso : Gráfico formato Gephi de un fichero perteneciente a una svn.

**1. Identificador.**

*WEBHF-01-dis-08*

**2. Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfWikiGephiServerRecursoGml.class
- Flujo de información servidor-cliente aceptable.
- Generación en formato correcto del fichero de datos con formato *gml* de Gephi.
- Comprobación de coherencia de los datos de origen con respecto a los gestionados.
- Visualización en el explorador a través del tipo de medio correcto (texto plano).

**3. Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*<http://localhost:9080/wiki/gephi/es.wikipedia.org/UCA/gml>*

**4. Especificación de la salida.**

Visualización de texto plano con formato *gml* de Gephi, de un artículo Mediawiki. Este texto plano se visualizará representando el flujo de información a través de líneas de texto que indican los nodos, Figura 5.17, y la relación entre estos a través de las aristas, Figura 5.18, en función de la última versión del artículo mediawiki.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización artículo wiki.



## 6. Requisitos especiales de procedimiento.

Ninguno.

## 7. Dependencias entre casos de prueba.

Este caso de prueba no depende de otro caso.

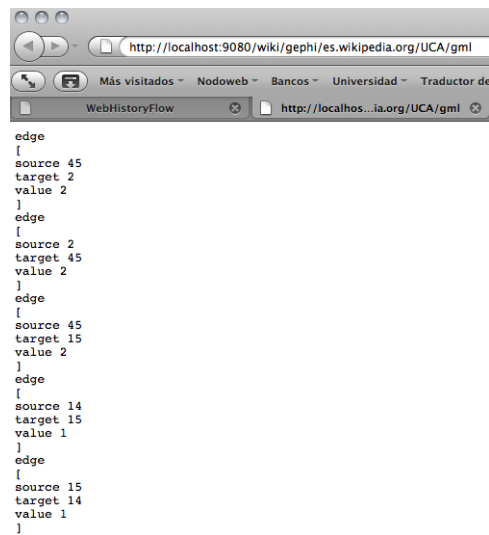


```

Creator "Francisco Jose Benitez Perez PFC-2011 WebHistoryFlow"
graph
{
  node
  {
    id 46
    label "ALLEBORGEBOT"
  }
  node
  {
    id 7
    label "LAUSEKER DIEGO"
  }
  node
  {
    id 1
    label "EMIJRP"
  }
  node
  {
    id 49
    label "VOLKOVBOT"
  }
  node
  {
    id 3
    label "XVAZQUEZ"
  }
  node
  {
    id 2
    label "TARAGUI"
  }
  node
  {
    id 32
    label "SENXX"
  }
}

```

Figura 5.17: Recurso : Fichero de datos de un artículo wiki con salida en formato Gephi, vista nodos.



```

edge
{
  source 45
  target 2
  value 2
}
edge
{
  source 2
  target 45
  value 2
}
edge
{
  source 45
  target 15
  value 2
}
edge
{
  source 14
  target 15
  value 1
}
edge
{
  source 15
  target 14
  value 1
}
}

```

Figura 5.18: Recurso : Fichero de datos de un artículo wiki con salida en formato Gephi, vista aristas.

**1. Identificador.**

*WEBHF-01-dis-09*

**2. Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfSvnGephiServerRecursoGml.class
- Ejecución correcta del parámetro *camino* que se especifica separado por puntos.
- Ejecución correcta en distintos servidores de subversiones.
- Flujo de información servidor-cliente aceptable.
- Generación en formato correcto del fichero de datos con formato *gml* de Gephi.
- Comprobación de coherencia de los datos de origen con respecto a los gestionados.
- Visualización en el explorador a través del tipo de medio correcto (texto plano).

**3. Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*http://localhost:9080/svn/gephi/svn.apache.org/  
repos.asf.subversion.trunk.notes/xfail-status/gml*

**4. Especificación de la salida.**

Visualización de texto plano con formato *gml* de Gephi, de un fichero perteneciente a una svn. Este texto plano se visualizará representando el flujo de información a través de líneas de texto que indican los nodos, Figura 5.19, y la relación entre estos a través de las aristas, Figura 5.20, en función de la última versión del artículo mediawiki.

**5. Necesidades de entorno.**

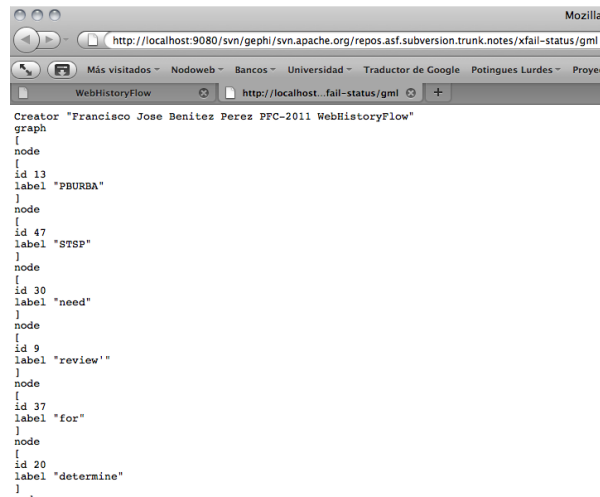
- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización de un fichero de svn.

**6. Requisitos especiales de procedimiento.**

Indicar el *camino* al fichero separado por puntos y no por barras.

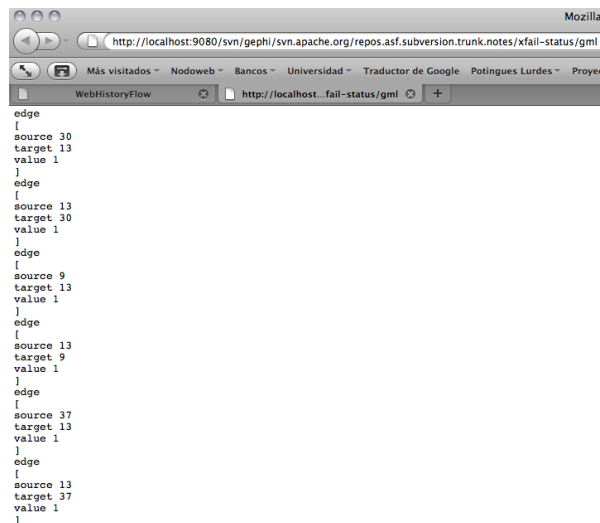
**7. Dependencias entre casos de prueba.**

Este caso de prueba no depende de otro caso.



```
Creator "Francisco Jose Benitez Perez PFC-2011 WebHistoryFlow"
graph
{
  node
  {
    id 13
    label "PBURBA"
  }
  node
  {
    id 47
    label "STSP"
  }
  node
  {
    id 30
    label "need"
  }
  node
  {
    id 9
    label "review"
  }
  node
  {
    id 37
    label "for"
  }
  node
  {
    id 20
    label "determine"
  }
}
```

Figura 5.19: Recurso : Fichero de datos de un fichero de una svn con salida en formato Gephi, vista nodos.



```
edge
{
  source 30
  target 13
  value 1
}
edge
{
  source 13
  target 30
  value 1
}
edge
{
  source 9
  target 13
  value 1
}
edge
{
  source 13
  target 9
  value 1
}
edge
{
  source 37
  target 13
  value 1
}
edge
{
  source 13
  target 37
  value 1
}
```

Figura 5.20: Recurso : Fichero de datos de un fichero de una svn con salida en formato Gephi, vista aristas.

**1. Identificador.**

*WEBHF-01-dis-10*

**2. Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfWikiPrefuseServerRecurso.class
- Visualización correcta de la representación de nodos y aristas en formato Prefuse a través de un *applet*.
- Flujo de información servidor-cliente aceptable.
- Generación en formato correcto del fichero temporal de datos con formato *XML* para su posterior uso en la demo applet de Prefuse.
- Comprobación de coherencia de los datos de origen con respecto a los gestionados.
- Similitud en cuanto al resultado del gráfico con respecto al gráfico en formato HF.
- Visualización de nodos en color amarillo representando lo(s) autores con mayor numero de palabras escritas.
- Visualización de nodos en color amarillo representado las palabras mas escritas por los autores.

**3. Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*<http://localhost:9080/wiki/prefuse/es.wikipedia.org/UCA/>*

**4. Especificación de la salida.**

Visualización del gráfico con formato demo Prefuse, de un artículo Mediawiki ,representando un flujo de información en forma de applet, a través de nodos y aristas en función de la última versión del artículo mediawiki como puede verse en la Figura 5.21, pudiendo controlar la visualización del applet a través de un panel de control, Figura 5.22.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización artículo wiki.
- Máquina Virtual de Java ver. 1.6

- 6. **Requisitos especiales de procedimiento.**  
La dirección URI indicada en el explorador web a de terminar con una barra /".
- 7. **Dependencias entre casos de prueba.**  
Este caso de prueba no depende de otro caso.

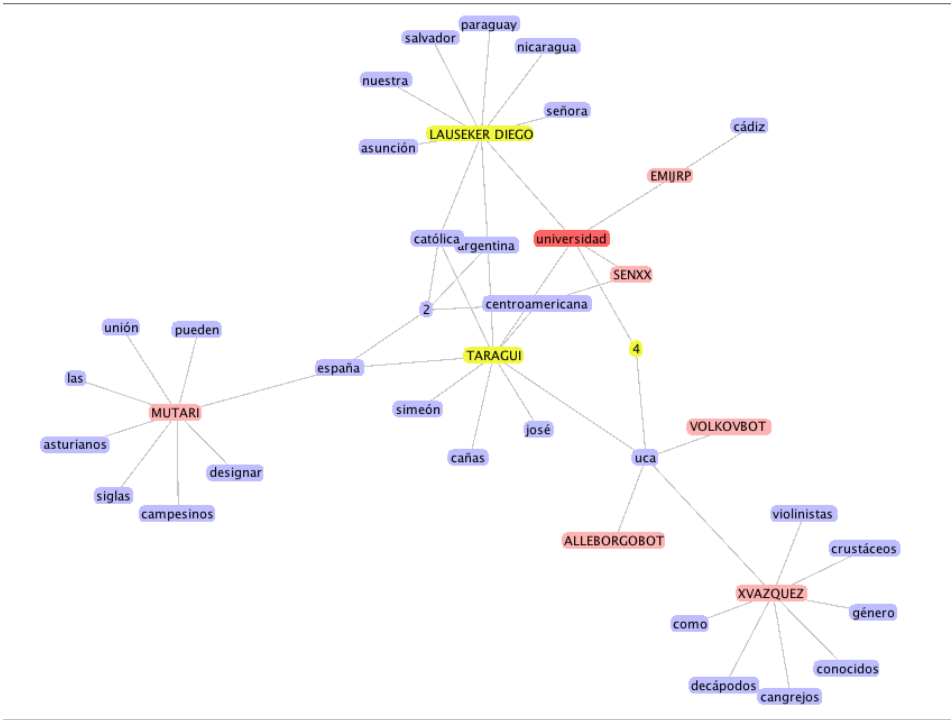


Figura 5.21: Recurso : Gráfico applet de artículo wiki con formato Prefuse.

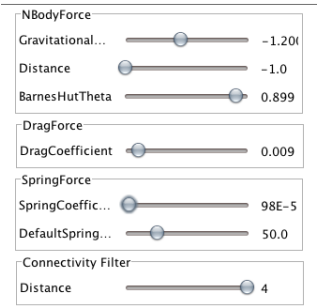


Figura 5.22: Recurso : Panel modificación vista del applet con formato Prefuse.

**1. Identificador.**

*WEBHF-01-dis-11*

**2. Elementos software que se van a probar.**

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfSvnPrefuseServerRecurso.class
- Visualización correcta de la representación de nodos y aristas en formato Prefuse a través de un *applet*.
- Flujo de información servidor-cliente aceptable.
- Generación en formato correcto del fichero temporal de datos con formato *XML* para su posterior uso en la demo applet de Prefuse.
- Comprobación de coherencia de los datos de origen con respecto a los gestionados.
- Similitud en cuanto al resultado del gráfico con respecto al gráfico en formato HF.
- Funcionamiento correcto en servidores de svn.
- Visualización de nodos en color amarillo representando lo(s) autores con mayor numero de palabras escritas.
- Visualización de nodos en color amarillo representado las palabras mas escritas por los autores.

**3. Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*[http://localhost:9080/svn/prefuse/svn.apache.org/  
repos.asf.subversion.trunk.notes/xfail-status/](http://localhost:9080/svn/prefuse/svn.apache.org/repos.asf.subversion.trunk.notes/xfail-status/)*

**4. Especificación de la salida.**

Visualización del gráfico con formato demo Prefuse, de un fichero de una subversión ,representando un flujo de información en forma de applet, a través de nodos y aristas en función de la última versión del artículo mediawiki como puede verse en la Figura 5.23, pudiendo controlar la visualización del applet a través del panel de control.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.

- Localización de un fichero de svn.
- Máquina Virtual de Java ver. 1.6

#### 6. Requisitos especiales de procedimiento.

La dirección URI indicada en el explorador web tiene que terminar con una barra "/".

#### 7. Dependencias entre casos de prueba.

Este caso de prueba no depende de otro caso.

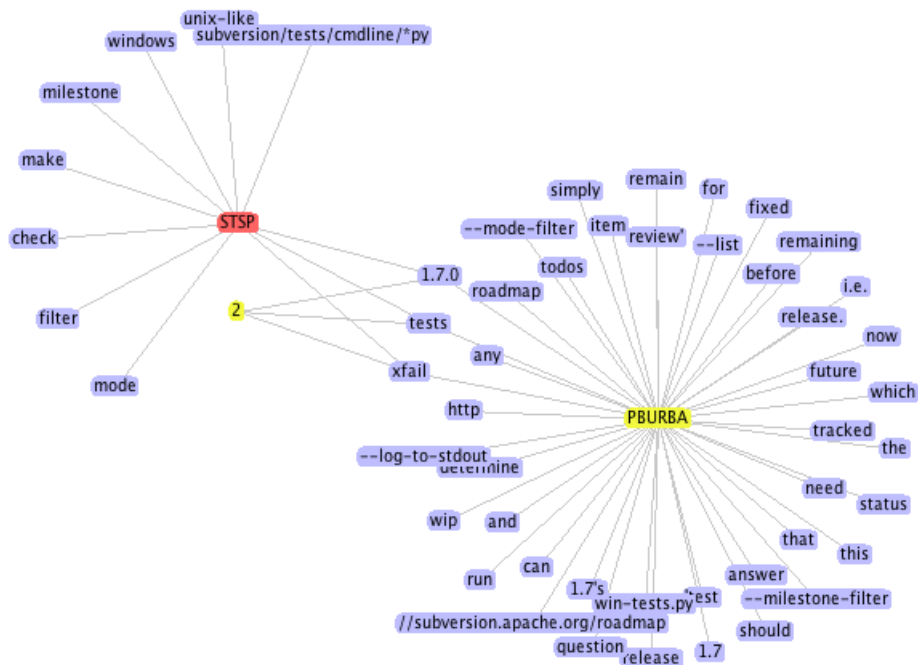


Figura 5.23: Recurso : Gráfico applet de fichero de una svn con formato Prefuse.

### 1. Identificador.

*WEBHF-01-dis-12*

### 2. Elementos software que se van a probar.

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfWikiPrefuseServerRecursoXml.class
- Flujo de información servidor-cliente aceptable.
- Generación en formato correcto del fichero de datos con formato *XML*.
- Comprobación de coherencia de los datos de origen con respecto a los gestionados.
- Visualización en el explorador a través del tipo de medio correcto (*XML*).

### 3. Especificación de la entrada.

Se indica en el explorador web la siguiente URI:

*http://localhost:9080/wiki/prefuse/es.wikipedia.org/UCA/xml*



Figura 5.24: Recurso : Gráfico fichero datos XML de un artículo wiki con formato Prefuse, vista nodos.

### 4. Especificación de la salida.

Visualización del fichero de datos con formato *XML* para Prefuse, de



un artículo mediawiki ,de un fichero de una svn ,formado por líneas de nodos, Figura 5.24, y aristas, Figura 5.25, en función de la última versión del artículo mediawiki.

#### 5. Necesidades de entorno.

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización artículo wiki.

#### 6. Requisitos especiales de procedimiento.

Ninguna.

#### 7. Dependencias entre casos de prueba.

Este caso de prueba no depende de otro caso.

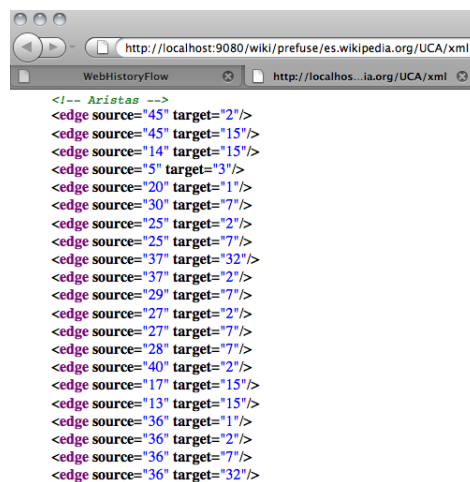


Figura 5.25: Recurso : Gráfico fichero datos XML de un artículo wiki con formato Prefuse, vista aristas.

### 1. Identificador.

*WEBHF-01-dis-13*

### 2. Elementos software que se van a probar.

- Elección correcta de la aplicación WebHfServerApplication del recurso WebHfSvnPrefuseServerRecursoXml.class
- Funcionamiento correcto en servidores de svn.
- Flujo de información servidor-cliente aceptable.
- Generación en formato correcto del fichero de datos *XML* de Prefuse.
- Comprobación de coherencia de los datos de origen con respecto a los gestionados.
- Visualización en el explorador a través del tipo de medio correcto (*XML*).

### 3. Especificación de la entrada.

Se indica en el explorador web la siguiente URI:

*http://localhost:9080/svn/prefuse/svn.apache.org/repos.asf.subversion.trunk.notes/xfail-status/xml*

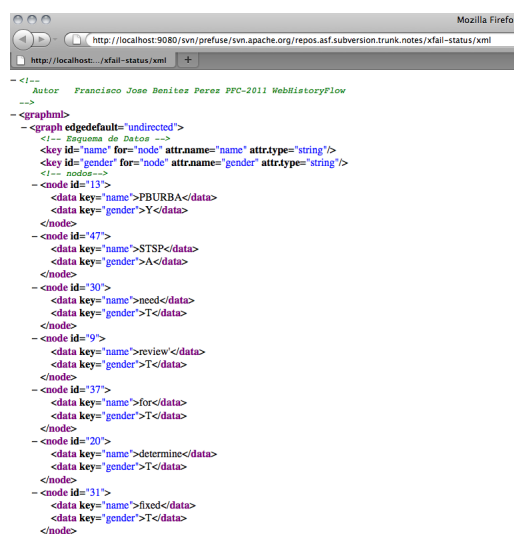


Figura 5.26: Recurso : Gráfico fichero datos XML formato Prefuse de fichero de una svn, vista nodos.

### 4. Especificación de la salida.

Visualización del fichero de datos con formato *XML* para Prefuse, de un fichero de una svn ,formado por líneas de nodos, Figura 5.26, y aristas, Figura 5.27, en función de la última versión del fichero.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización fichero de una svn.

**6. Requisitos especiales de procedimiento.**

Ninguna.

**7. Dependencias entre casos de prueba.**

Este caso de prueba no depende de otro caso.

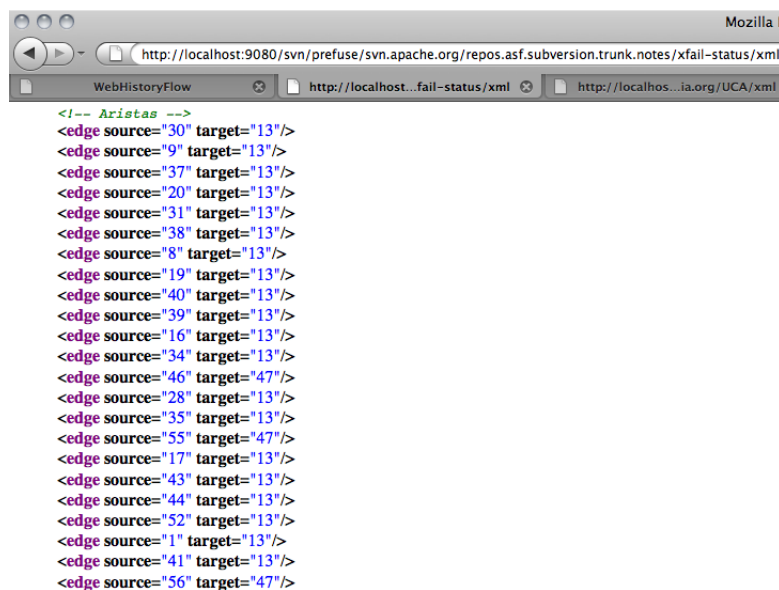


Figura 5.27: Recurso : Gráfico fichero datos XML formato Prefuse de fichero de una svn, vista aristas.

1. **Identificador.**

*WEBHF-01-dis-14*

2. **Elementos software que se van a probar.**

- Ejecutando recurso en modo : *individual* tanto para artículo wiki como para fichero de svn, *sin especificar*, un autor determinado.

3. **Especificación de la entrada.**

Se indica en el explorador web la siguiente URI:

*http://localhost:9080/wiki/es.wikipedia.org/UCA/individual*

4. **Especificación de la salida.**

Visualización del mensaje como tipo de medio en formato texto plano, indicando que no se ha indicado el autor, como muestra la Figura 5.28.

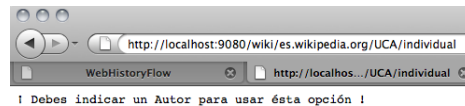


Figura 5.28: Recurso : Mensaje error en modo: *individual*, sin indicar autor.

5. **Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización artículo wiki o fichero svn.

6. **Requisitos especiales de procedimiento.**

Ninguno.

7. **Dependencias entre casos de prueba.**

Este caso de prueba no depende de otro caso.

**1. Identificador.**

*WEBHF-01-dis-15*

**2. Elementos software que se van a probar.**

- Especificar una dirección URI que *no coincida* con ninguna de las rutas establecidas para la ejecución de los recursos disponibles.
- Indicar modos no existentes.

**3. Especificación de la entrada.**

Se indica en el explorador web URI's como :

*[http://localhost:9080/no\\_existe/es.wikipedia.org/UCA/grupo](http://localhost:9080/no_existe/es.wikipedia.org/UCA/grupo) ,*

*[http://localhost:9080/wiki/gephi/es.wikipedia.org/UCA/no\\_existe](http://localhost:9080/wiki/gephi/es.wikipedia.org/UCA/no_existe)*

**4. Especificación de la salida.**

Visualización del mensaje de error por parte del framework de Restlet indicando que la ruta no existe, Figura 5.29, o el mensaje en texto plano indicando que alguno de los parámetros indicados no es correcto, Figura 5.29.

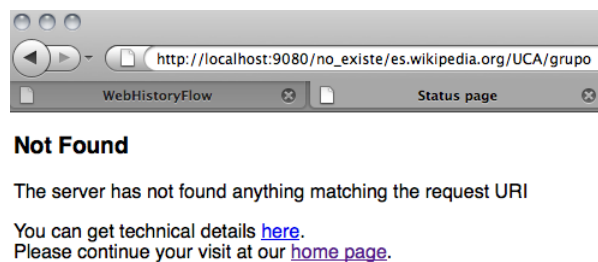


Figura 5.29: Recurso : Mensaje error Restlet cuando no encuentra coincidencias en rutas.

**5. Necesidades de entorno.**

- Ordenador con Sistema Operativo MacOSx, Linux o Windows.
- IDE Netbeans.
- Explorador web.
- Conexión a internet.
- Servidor Tomcat en Apache.
- Aplicación WebHistoryFlow en ejecución.
- Localización artículo wiki o fichero svn.

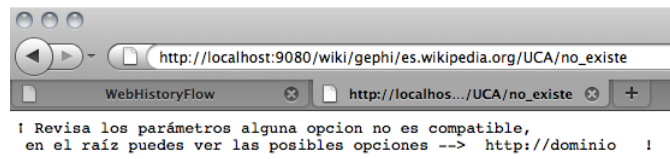


Figura 5.30: Recurso : Mensaje error cuando no existe la opción en las rutas definidas.

**6. Requisitos especiales de procedimiento.**

Ninguno.

**7. Dependencias entre casos de prueba.**

Este caso de prueba no depende de otro caso.

**5.3.1.4. Especificación de procedimientos de prueba****1. Identificador del procedimiento de prueba correspondiente al diseño de prueba.**

*WEBHF-01*

**2. Objetivo del procedimiento y lista de casos que se ejecutan con él.**

A partir de este documento de procedimientos de prueba se especifican los pasos para la ejecución del conjunto de casos de pruebas utilizadas para analizar los elementos software.

Las referencias de los casos de pruebas comprobados y completados pertenecen al siguiente rango : [*WEBHF-01-dis-01 al WEBHF-01-dis-15*]

**3. Requisitos especiales para la ejecución.**

- A la hora de indicar el camino para un fichero de una subversión, los distintos subdirectorios se separarán con un punto.
- Para solicitar recursos gráficos en formato Prefuse, la URI, ha de terminar en barra "/"

**4. Pasos en el procedimiento:**

- Secuencia necesaria de acciones para preparar la ejecución.
  - Localizar en la *www*, un artículo MediaWiki o un fichero de una *svn*.
  - Escribir en el explorador la URI, en modo correcto, del artículo o fichero a solicitar.
- **Acciones necesarias para empezar la ejecución.**
  - Lado del servidor : Aplicación WebHistoryFlow ejecutándose.
  - Lado del cliente : Abrir explorador web e introducir URI correcta.
- **Acciones necesarias durante la ejecución.**

Ninguna.

**5. Cómo se realizarán las medidas.(tiempo de respuesta).**

PENDIENTE

**6. Acciones necesarias para suspender la prueba.**

Pulsar cancelar en el explorador o detener la aplicación en el servidor Tomcat de Apache.

**7. Acciones necesarias para tratar los acontecimientos anómalos.**

- Parar y reiniciar la aplicación en el servidor Tomcat de Apache.
- Vaciar la memoria cache del explorador web si está involucrado visualizaciones de *applets* desde las opciones de *Prefuse*.

## 5.4. Utilización de recursos WebHistoryFlow en implementaciones cliente

Como ya se comentó en apartados anteriores la principal misión de la aplicación WebHistoryFlow, como su nombre de proyecto indica es, establecer un *servicio web para la visualización gráfica de las contribuciones en webs colaborativas*, y para constatar la utilidad de la misma en actividades reales en un entorno web se describen algunos ejemplos de utilización a partir de posibles implementaciones que se realizaría desde el lado del cliente.

- Se podrían realizar llamadas desde Gestores de Contenidos como *Moodle*, *Joomla* etc..., como podemos ver en la Figura 5.31 con Joomla. Para complementar la información de texto se añaden gráficos que son incrustados a través de llamada(s) URI a uno de los recursos de WebHistoryFlow. Uno de los aspectos mas interesantes, a parte de mostrar la situación gráfica en ese momento del flujo de información de un artículo wiki o un fichero de una subversión, es que el gráfico no sería estático ya que cada vez que se hace la llamada al recurso se calcula la situación actual del artículo o fichero de la subversión, con lo cual tendríamos un gráfico dinámico.

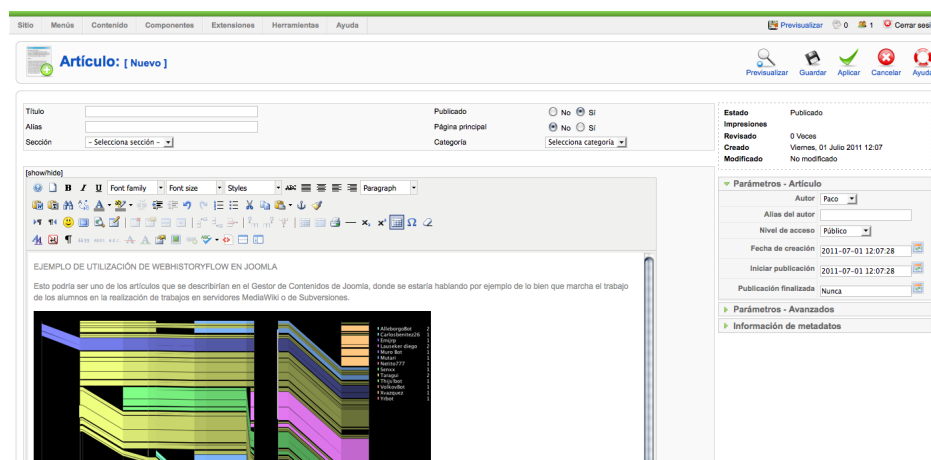


Figura 5.31: Gestor de Contenidos Joomla solicitando recurso gráfico a WebHistoryFlow.

- Podríamos implementar una página web, e incluir las llamadas que queramos para tener dicha web con una representación gráfica actualizada y dinámica, Figura 5.32.

Para mostrar como sería el código a incluir desde HTML en una página web, muestro en la Figura 5.33, como sería. Por simplificar he editado el que proporciona el editor de html de Joomla.



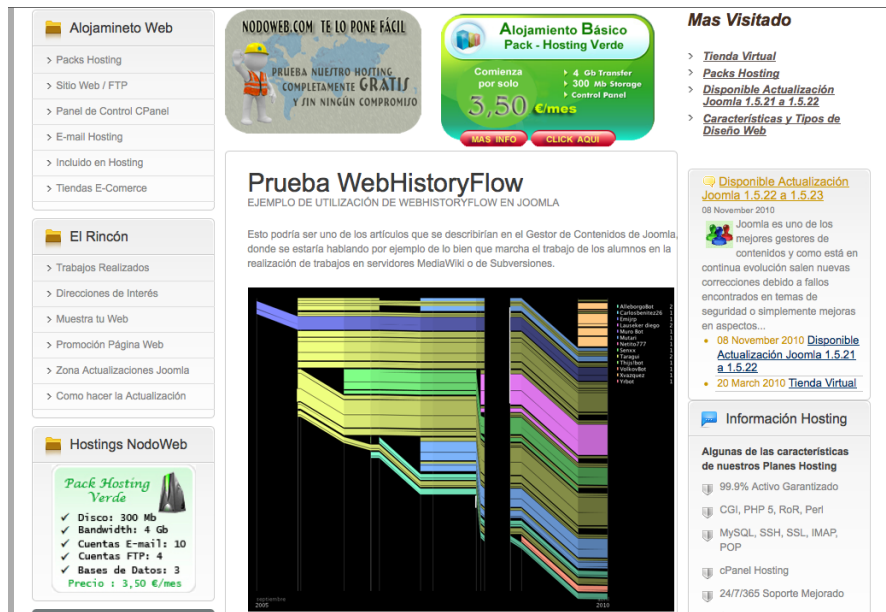


Figura 5.32: Uso de WebHistoryFlow en una página web de Joomla.

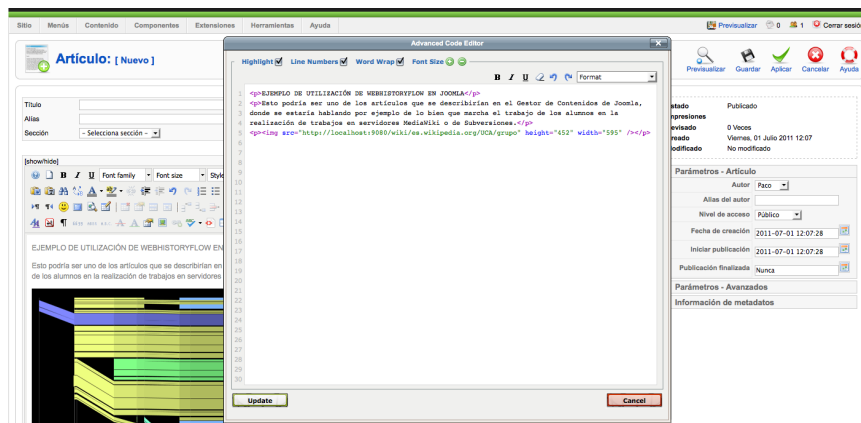


Figura 5.33: Gestor de Contenidos Joomla, edición de código HTML del artículo.

- Se podría realizar llamada(s) desde cualquier lenguaje de programación (Java, Php, Perl etc...) para solicitar recurso y servirlos a otros clientes o mostrarlo como parte de una aplicación de escritorio como se muestra en la Figura 5.34. Para ello he implementado una pequeño script en Java a partir de un JFrame, Figura 5.35.

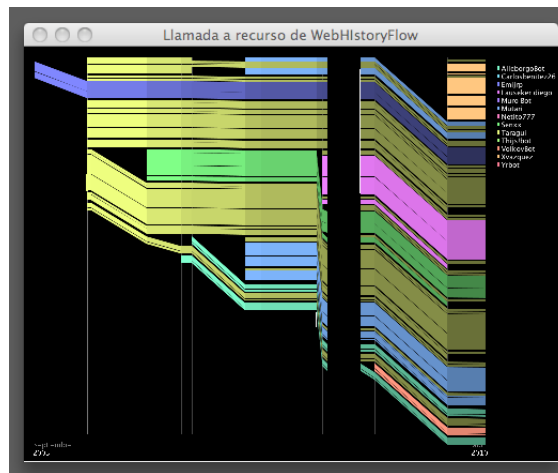


Figura 5.34: Ventana JFrame de Java con recurso WebHistoryFlow.

```

1 package webhistoryflowprueba;
2
3 import java.awt.*;
4 import java.awt.image.*;
5 import java.io.*;
6 import javax.swing.*;
7 import java.net.URL;
8 import javax.imageio.ImageIO;
9
10 public class WebHistoryFlowPrueba extends JFrame {
11
12     public WebHistoryFlowPrueba() {
13         super("Llamada a recurso de WebHistoryFlow");
14     }
15
16     @Override
17     public void paint(Graphics g) {
18         Graphics2D g2 = (Graphics2D)g;
19         try {
20             URL url = new URL("http://localhost:9080/wiki/es.wikipedia.org/UCA/grupo");
21             BufferedImage imagen = ImageIO.read(url);
22             int ancho = imagen.getWidth()/2;
23             int alto = imagen.getHeight()/2;
24             g2.drawImage(imagen, 0, 20, ancho, alto, this);
25         } catch (IOException e) {}
26     }
27
28     public static void main(String args[]) {
29         WebHistoryFlowPrueba recurso = new WebHistoryFlowPrueba();
30         recurso.setDefaultCloseOperation(EXIT_ON_CLOSE);
31         recurso.setSize(500,410);
32         recurso.setResizable(false);
33         boolean ahorasi = true;
34         /* si se cumple la condición mostraría una ventana con la
35          * solicitud del recurso a WebHistoryFlow
36          * http://localhost:9080/wiki/es.wikipedia.org/UCA/grupo
37          */
38         if(ahorasi)
39             recurso.setVisible(true);
40     }
41 }
42
43
44

```

Figura 5.35: Ejemplo implementación Java para uso de WebHistoryFlow.

- Se pueden solicitar recursos de datos en formato Gephi para su posterior uso en la *aplicación de escritorio* de *Gephi*, Figuras 5.36, 5.37, con lo que obtendríamos distintos tipos de representaciones.

## 5.4. UTILIZACIÓN DE RECURSOS WEBHISTORYFLOY EN IMPLEMENTACIONES CLIENTE

153

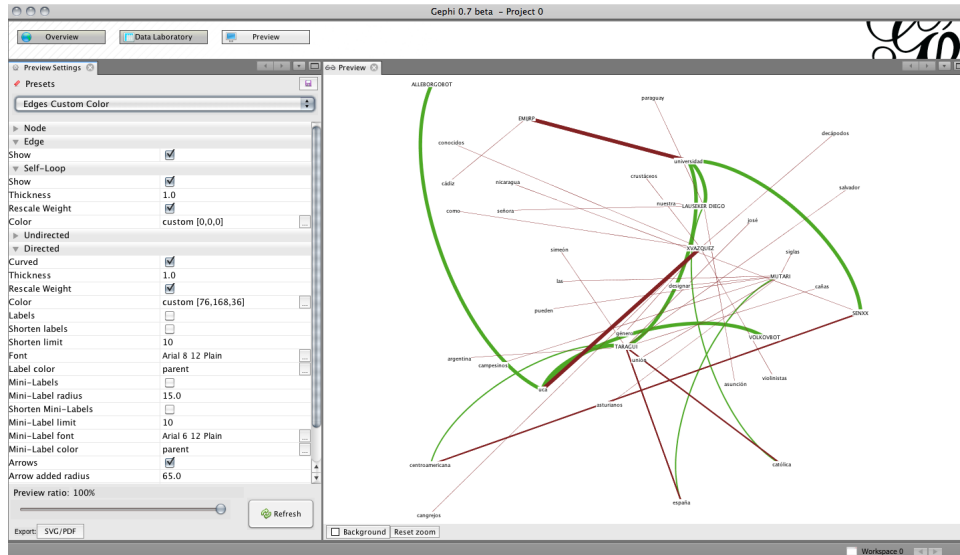


Figura 5.36: Uso de fichero *.gml* creado con WebHistoryFlow en Aplicación Escritorio Gephi 7.0

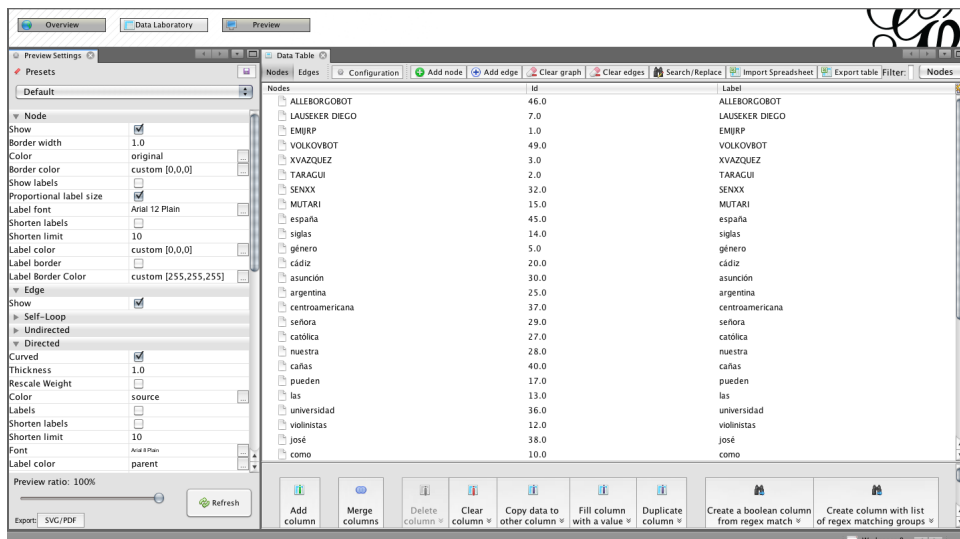


Figura 5.37: Datos *.gml* creado con WebHistoryFlow en Aplicación Escritorio Gephi 7.0



## Capítulo 6

# Conclusiones

En la mayoría de los trabajos que se realizan en la vida laboral de cualquier trabajador, uno de los aspectos mas incómodos es realizar un determinado trabajo a partir de algo que ya está construido, ¿porqué?, pues porque además de tener que hacer o conseguir un objetivo concreto, previamente hay que entender detalladamente como es el funcionamiento de lo que estás tratando además de manipularlo y adaptarlo al proceso o tarea que tienes que realizar.

En mi caso el proyecto presentado ha consistido en un trabajo que descansa en una aplicación ya construida y por lo tanto con los inconvenientes anteriormente expuestos.

Después de varios meses trabajando en el análisis, diseño, implementación y pruebas de la aplicación WebHistoryFlow puedo llegar a las siguientes conclusiones:

- El objetivo general que se buscaba era encapsular una aplicación de escritorio para su versionado en modo online y tras muchas horas de análisis de como estaba construida la aplicación original se pudo extraer los puntos importantes que realmente se usarían para su posterior adaptación y con ello su encapsulación.

Para poder realizar el encapsulado y poder llegar a crear la aplicación WebHistoryFlow online, he encontrado algunos problemas interesantes como:

- El código fuente de la aplicación original History Flow no estaba disponible por parte de sus autores y sólo permitían la descarga de:
  - Una aplicación de escritorio descargable para usar con el plugin para artículos mediawiki (disponible en su web [\[20\]](#) ).
  - Un fichero *.jar* para aquellas personas que quisieran construir plugins acoplables a su aplicación y así poder aplicar distintas formas de entrada de datos.

Lógicamente a la hora de trabajar con el fichero .jar en el IDE de Netbeans, este fichero solo contiene paquetes ya compilados que no se pueden usar para su manipulación, por lo tanto solo es usado para la ejecución de clases de importancia secundaria y que no necesitaban modificarse. Todas aquellas clases que han exigido una modificación, han sido procesadas con decompiladores.

Para poder trabajar con el código fuente de la aplicación se tuvo que aplicar varios decompiladores para poder obtener dicho código. Además otro inconveniente que suele aparecer en los decompiladores es que no todo lo que decompilan suele ser correcto, es decir, casi seguro que se cuelan varias líneas de código que no tienen sentido y si para colmo tienes la suerte que estas líneas de código sin sentido, coinciden en las clases que son fundamentales para la ejecución de la aplicación, pues es un gran problema añadido.

Desafortunadamente me coincidieron líneas incongruentes en clases fundamentales y tras varios días de análisis del funcionamiento de todas las clases fundamentales que daban problemas se insertó el código que supuestamente necesitaba para su ejecución. Finalmente se implementó el código que faltaba en las distintas clases y se consiguió un funcionamiento primitivo de la función base de la aplicación, es decir, mostrar un gráfico acorde a los datos xml de un artículo wiki.

- La arquitectura de construcción original no era la adecuada para poder gestionar distintos plugins y recursos. Para ello hubo que decodificar muchas clases y adaptarlas al nuevo diseño. Finalmente se consiguió un nuevo esquema que sí era compatible con la forma buscada, solicitud-respuesta.
- Además del objetivo general hay que describir algunos objetivos complementarios, y no por ello menos importantes, donde se llegan a las siguientes conclusiones:
  - Para conseguir que la aplicación funcionase en modo concurrente, además de trabajar con Java que solucionaba este inconveniente casi solo a través de hilos, se hizo necesario la utilización del framework de Restlet que cumplió con creces las expectativas esperadas, facilitando las tareas de concurrencia y acceso a recursos. Con Restlet, entre otras cosas, se buscó y se consiguió una metodología de trabajo basada en una arquitectura REST.
  - Se pudo obtener datos a partir de los plugin de HistoryFlow para importar datos xml de un artículo mediawiki y del plugin de Richard Feard para datos de un fichero de una subversión. Previamente hubo que modificar un poco la forma de acceder a estos dos

plugins, creando una clase específica para la preparación de datos y redirección correcta al plugin a usar , así como prescindir de clases, existentes en los plugins , que no eran necesarias para la aplicación a construir ya que tocaban aspectos relacionados con la interfaz de usuario exigida en la aplicación original.

- Gracias a Restlet se pudo aplicar correctamente el enrutado de los distintos recursos de una manera sencilla y eficaz. Permitiendo además que cada parámetro enviado en la URI fuese interpretado no como la ejecución típica de peticiones de órdenes, Buscar ?nom=Pedro, MostrarGraf ? tipo=Prefuse ... modo verbo(buscar, mostrar...), sino que se han realizado las peticiones como si fueran Sustantivos(Autor artículo Prefuse, Gráfico de un grupo de Autores,...), por ejemplo, Gráfico en Grupo Artículo UCA, Gráfico Prefuse de un Autor, es decir no se han realizado peticiones directas a funciones o métodos, sino que se han exigido los datos o gráficos de algo o alguien directamente.
- Al tratar con distintas tecnologías fue necesario disponer de un modo de trabajo que diera salida a estos medios. Restlet fue la solución correcta para ello ya que respondía al cliente a través de los distintos tipos de medio que tiene disponible su framework. Se pudo realizar correctamente la devolución del tipo de medio(PDF, XML, TXT, HTML, PNG), en función del recurso solicitado por el cliente, gracias al manejador disponible en Restlet y la clase-método usada para ello (response.setEntity).
- Por ultimo podíamos decir que de todo este proceso de gestación y creación de la aplicación he asimilado algunas cosillas interesantes a nivel personal y profesional como:
  - Un mejor conocimiento del lenguaje de programación en Java, que sin llegar a ser un profesional, he conseguido tocar muchos aspectos de Java que espero me sirvan en un futuro.
  - Cómo adaptar una aplicación a una arquitectura de trabajo como la disponible en el framework de Restlet.
  - Manejo a fondo o superficialmente de otras aplicaciones conocidas o nuevas como Netbeans, gestores de subversiones, aplicaciones de clasificación de referencias, aplicaciones gráficas de escritorio como las de HistoryFlow y Gephi, aplicaciones para gestión de contenidos en subversiones, aplicaciones gráficas para diagramas, aplicaciones para Latex, etc...
  - Que el trabajo sin una motivación no ayuda a conseguir objetivos y que los resultados dependerán de tu esfuerzo personal.

- A realizar una distribución realista del trabajo pendiente, sin prisas pero sin pausas.
- A confirmar que la constancia en el trabajo es la clave fundamental para la consecución de algo, y me consta que esa ha sido mi actitud desde que empecé el proyecto. Como se dice popularmente *El que la sigue la consigue*.
- He preparado, diseñado, adaptado y ejecutado con éxito, en mayor o menor grado, una aplicación y una memoria que juntos forma *El Proyecto*.

Aunque el alcance del proyecto esta inicialmente acotado, la solución propuesta en este *Servicio Web*, es flexible, extensible y suficientemente comentada como para que en futuros trabajos se pueda ampliar con otros orígenes de datos y motores de visualización gráfica.

Como conclusión final decir que todos los objetivos marcados inicialmente han sido cumplidos con creces e incluso alguno de ellos mejor de lo que inicialmente me esperaba, y por ello puedo decir que me siento satisfecho del trabajo realizado.



## Capítulo 7

# Apéndice A

### 7.1. Definiciones, acrónimos y abreviaturas

**HF** Siglas para definir la aplicación History Flow.

**IEEE** Corresponde a las siglas de (Institute of Electrical and Electronics Engineers) en español Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.

**CU** Caso de Uso.

**CLAP** Acrónimo de Protocolo de Acceso al Cargador de Clases.

**DSS** Diagrama de Secuencia de Sistemas para procesos con UML.

**UML** Lenguaje de Modelado Unificado.

**Subversión** Subversión es un sistema de control de versiones.

**Control de versiones** control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

**MoinMoin** Es un motor wiki avanzado, fácil de utilizar, y extensible.

**Wiki** Sitios webs que pueden ser modificados libremente por los usuarios que las visitan.

**GML** Geography Markup Language, es un lenguaje estándar de codificación, es una gramática XML para expresar las características geográficas.

**JSP** JavaServer Page. Genera contenido dinámico en un entorno web.

**Socket** Conexión de red.

**InputStreamReader** Clase Java que recibe un flujo de datos de entrada. Usada por el analizador sintáctico para procesar dicho flujo de datos.

**API** Interfaz de Programación de Aplicaciones. Es un conjunto de rutinas que proporciona una aplicación o una biblioteca (conjunto de clases agrupadas en un conjunto llamado API).

**Frameworks** Es una plataforma, entorno o marco de trabajo, es una estructura de soporte donde un proyecto software se puede organizar o desarrollar.

**URL** Localizador de recursos uniforme. La URL no define por sí misma cual es el contenido al que se refiere solo indica donde y como localizar el fichero. La URL está compuesto por cuatro partes:

- **Protocolo** para acceder al archivo
- **Nombre computadora.** Normalmente un dominio o una dirección IP.
- **Directorios.** Camino de directorios para acceder al fichero separados por ('/').
- **Archivo** donde se encuentra el recurso solicitado.

**EJB** Enterprise JavaBeans. Son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE (ahora JEE 5.0) de Oracle Corporation (inicialmente desarrollado por Sun Microsystems).

**WADL** Lenguaje de Descripción de Aplicaciones Web. Es un formato de archivo basado en XML que proporcionan una descripción legible de aplicaciones web, basado en HTTP.

**Proxy** Un proxy permite a otros equipos conectarse a una red de forma indirecta a través de él. Cuando un equipo de la red desea acceder a una información o recurso, es realmente el proxy quien realiza la comunicación y a continuación traslada el resultado al equipo inicial.

**Idempotente** Indica que múltiples peticiones idénticas proporcionan una misma solución. Ejemplo borrar dos veces lo mismo.

**Parsear** Proceso de analizar una secuencia de símbolos a fin de determinar su estructura gramatical con respecto a una gramática formal dada. Formalmente es llamado análisis de sintaxis. Un parseador (parser) es un programa de computación que lleva a cabo esta tarea.

**JAXP** API Java para Proceso de XML facilita el proceso de datos XML para aplicaciones escritas en lenguaje Java. Contiene analizadores estándar como SAX y DOM.

**SAX** API Simple para XML.

**DOM** Modelo en Objetos para la representación de Documentos.

**Ruby on Rails** Framework de aplicaciones web de código abierto.

**IDE** Entorno de Desarrollo Integrado.

**GPL** Licencia Pública General.

**BSD** Distribución de software berkeley.

**Servlet** Programa que se ejecuta en el servidor.

**Applet** Programa que se ejecuta en el contexto del navegador web.

**CPAL** Licencia versión 1.0 de Reconocimiento Común Público.

**JVM** Máquina Virtual de Java es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

**Servidor Web** Es un ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras.

**Apache** Servidor web de distribución libre y de código abierto.

**Tomcat** Tomcat es un servidor web con soporte de servlets y JSPs. Escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

**Renderización** Proceso para generar una imagen desde un modelo.

**HTTP** Protocolo cliente-servidor. Un cliente envía un mensaje de petición a un servidor, el cuál, después de realizar algún tipo de procesamiento, devuelve un mensaje de respuesta que contiene un código de estado y el resultado de la petición.

**WebHFlow** Aplicación software para uso On-line. Suministra recursos a partir de una petición URI.

**REST** Transferencia de Estado Representacional. Es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.

**Actor** Es una idealización de una persona externa, de un proceso, o de una cosa que interactúa con un sistema, un subsistema, o una clase.

**Caso de uso** CU Es una unidad coherente de funcionalidad, externamente visible, proporcionada por una unidad del sistema y expresada por secuencias de mensajes intercambiados por la unidad del sistema y uno o más actores.

**SVNKit** Herramienta Java de código fuente abierto, cumple todas las características de subversión y proporciona API's para trabajar con subversiones, acceder a ellas y manipular los repositorios de subversión, todo dentro de la aplicación Java.

**www** World Wide Web es un sistema de distribución de información basado en hipertexto o hipermedios enlazados.

**Joomla** Es un sistema de gestión de contenidos, y entre sus principales virtudes está la de permitir editar el contenido de un sitio web de manera sencilla. Aplicación de código abierto programada mayoritariamente en PHP bajo una licencia GPL.

**URI** Identificador uniforme de recurso. Es una cadena de caracteres que identifica inequívocamente un recurso.

## Capítulo 8

# Apéndice B

### 8.1. Guía de usuario

Para facilitar la utilización de la aplicación a continuación se describe una pequeña guía de usuario de la aplicación WebHistoyFlow. Además se indica una serie de links para facilitar al usuario su forma de uso.

A continuación describo algunas indicaciones a tener en cuenta a la hora de escribir la dirección URI del recurso a solicitar:

- Solo en el caso de solicitar recursos relacionados con el gráfico en formato Prefuse, habrá que indicar la barra inclinada, / , al final de la URI.  
Ejem. <http://localhost:8115/wiki/prefuse/es.wikipedia.org/UCA/>
- Cuando se solicite cualquier recurso relacionado con ficheros pertenecientes a subversiones hay que tener en cuenta dos aspectos :
  1. El dominio del *host* donde se encuentra el fichero para analizar.  
Ejemplo:  
<http://localhost:8115/svn/gephi/svn.apache.org/repos.asf.subversion.trunk.notes/xfail-status/gml>
  2. El camino de directorios hasta donde se encuentra dicho fichero.  
Ejemplo:  
<http://localhost:8115/svn/gephi/svn.apache.org/repos.asf.subversion.trunk.notes/xfail-status/gml>
- El camino del fichero, al solicitar recursos relacionados con subversiones, estarán separados por puntos.
- La separación entre el dominio del *host* y el *camino* estará separada por la barra inclinada / .

UCA - PFC 2011 WebHistoryFlow	Formato para Wikis
Gráficos HistoryFlow	host_server/wiki/host_wiki/articulo/modo
Gráficos HistoryFlow con Autor	host_server/wiki/host_wiki/articulo/individual/autor
Gráficos Gephi	host_server/wiki/gephi/host_wiki/articulo
Salida fichero GML - Gephi	host_server/wiki/gephi/host_wiki/articulo/gml
Gráficos Applet nodos-aristas Prefuse	host_server/wiki/prefuse/host_wiki/articulo/
Salida fichero XML - Prefuse	host_server/wiki/prefuse/host_wiki/articulo/xml

Tabla 8.1: Formatos predefinidos en artículos mediawiki para WebHistoryFlow.

UCA - PFC 2011 WebHistoryFlow	Ejemplo para Wikis
Gráficos HistoryFlow	./wiki/es.wikipedia.org/UCA/grupo
Gráficos HistoryFlow con Autor	./wiki/es.wikipedia.org/UCA/individual/Emijrp
Gráficos Gephi	./wiki/gephi/es.wikipedia.org/UCA
Salida fichero GML - Gephi	./wiki/gephi/es.wikipedia.org/UCA/gml
Gráficos Applet nodos-aristas Prefuse	./wiki/prefuse/es.wikipedia.org/UCA/
Salida fichero XML - Prefuse	./wiki/prefuse/es.wikipedia.org/UCA/xml

Tabla 8.2: Ejemplos solicitud recursos WebHistoryFlow para artículos mediawiki.

Para que el usuario web pueda realizar algunas pruebas de ejemplo con la aplicación WebHistoryFlow a continuación se representan los formatos de uso y algunos ejemplos.

En la Tabla 8.1 se puede ver cuales son los formatos predefinidos para artículos mediawiki.

En la Tabla 8.2 se proporcionan algunos ejemplos de llamadas a recursos a partir de artículos mediawiki.

En la Tabla 8.3 se puede ver cuales son los formatos predefinidos para

UCA - PFC 2011 WebHistoryFlow	Formato para Ficheros de Svn
Gráficos HistoryFlow	host_server/svn/ <b>host_svn</b> /camino/fichero/modo
Gráficos HistoryFlow con Autor	host_server/svn/ <b>host_svn</b> /camino/fichero/individual/autor
Gráficos Gephi	host_server/svn/gephi/ <b>host_svn</b> /camino/fichero
Salida fichero GML - Gephi	host_server/svn/gephi/ <b>host_svn</b> /camino/fichero/gml
Gráficos Applet nodos-aristas Prefuse	host_server/svn/prefuse/ <b>host_svn</b> /camino/fichero/
Salida fichero XML - Prefuse	host_server/svn/prefuse/ <b>host_svn</b> /camino/fichero/xml

Tabla 8.3: Formatos predefinidos para ficheros subversiones en WebHistoryFlow.

ficheros pertenecientes a subversiones.

En la Tabla 8.4 se proporcionan algunos ejemplos de llamadas a recursos a partir de ficheros de una subversión.

A continuación se explican las palabras resaltadas en negrita con los distintos parámetros disponibles.

- **host\_wiki**. Host de artículos tipo mediawiki, como :
  - es.wikipedia.org
  - meta.wikimedia.org
  - www.mediawiki.org
- **host\_svn**. Host de las svn que contiene el fichero, como :
  - svn.apache.org
- **artículo**. Nombre del artículo wiki.
- **fichero**. Nombre del fichero de una svn que queremos analizar.
- **modo**. Define el modo de visualización del gráfico WebHistoryFlow. Hay predefinidos cinco modos de visualización :
  - **grupo** : Representa las contribuciones de todos los autores a través de colores.
  - **tiempo** : Representa la persistencia de las contribuciones a través del tiempo.

<b>UCA - PFC 2011 WebHistoryFlow</b>
<b>Ejemplo para Ficheros de Svn</b>
<b>Gráficos HistoryFlow</b>
./svn/subversion.assembla.com/svn.iff6.trunk.Akavriumas/ Akvariumas Model.emx/grupo
<b>Gráficos HistoryFlow con Autor</b>
./svn/subversion.assembla.com/svn.iff6.trunk.Akavriumas/ Akvariumas Model.emx/individual/deividas
<b>Gráficos Gephi</b>
./svn/gephi/svn.apache.org/repos.asf.subversion.trunk.notes/xfail-status
<b>Salida fichero GML - Gephi</b>
./svn/gephi/svn.apache.org/repos.asf.subversion.trunk.notes/xfail-status/gml
<b>Gráficos Applet</b>
<b>nodos-aristas Prefuse</b>
./svn/prefuse/svn.apache.org/repos.asf.subversion.trunk.notes/xfail-status/
<b>Salida fichero</b>
<b>XML - Prefuse</b>
./svn/prefuse/svn.apache.org/repos.asf.subversion.trunk.notes/xfail-status/xml

Tabla 8.4: Ejemplos solicitud recursos WebHistoryFlow para ficheros de subversiones.

- **link** : Muestra la colocación de los links sobre las páginas y su persistencia a lo largo del tiempo.
  - **reciente** : Resalta la porción correspondiente a los textos incluidos recientemente en cada versión.
  - **individual** : Resalta la contribución de un autor a traves de todas las versiones.
- **autor** : Nombre de uno de los autores que pertenecen a esa wiki o participan en un fichero de una svn.

Las descripciones incluidas en los laterales de las tablas están formado por los siguientes recursos que se pueden solicitar :

- **Gráficos HistoryFlow** : Visualiza gráficos en formato HistoryFlow, representando el flujo de vida de una wiki o una svn, así como los autores que participan en dicha wiki o svn, clasificado por colores.
- **Gráficos HistoryFlow con Autor** : Idem a la anterior pero con la posibilidad de indicar un autor concreto de los participantes en esa wiki o fichero de la svn, resaltando su aportación durante todas las versiones.



- **Gráficos Gephi** : Devuelve un gráfico en formato PDF, a partir de una implementación demo, descargable desde la web de Gephi, pero a través de los datos proporcionados por WebHistoryFlow.
- **Salida fichero GML - Gephi** : Suministra un fichero en formato *.gml* que es uno de los utilizados por la API de Gephi. Este fichero puedes cargarlo en la aplicación de escritorio suministrada en su web y realizar el tipo de gráfico que mas te guste.
- **Gráficos Applet nodos-aristas Prefuse** : Visualiza un applet en el cual se visualiza una representación gráfica a partir de nodos y aristas de la última versión de una wiki o una svn, mostrando la relación entre los autores de esa versión y las palabras escritas por ellos, así como el autor(es) que mas palabras proporcionan y las palabras que mas se repiten. Estas dos últimas características se representan con nodos de color amarillo. Una característica de esta demo es que al posicionar el ratón sobre cualquier nodo se activan los que están relacionados.
- **Salida fichero XML - Prefuse** : Devuelve un fichero en formato *.xml* en el cual se puede observar los distintos nodos que representan dicha versión de wiki o svn, así como las aristas que los unen. Igualmente puede ser usado para usar en alguna implementación gráfica de la API Prefuse.



# Bibliografía

- [1] Booch, G; Jacobson, I.; Rumbaugh, J.M.; *UML. El lenguaje unificado de modelado. Manual de Referencia*, Ed. Addison Wesley, 2000.
- [2] Raúl Alarcón, *Diseño orientado a objetos con UML*, Grupo EIDOS Consultaría y Documentación Informática, S.L., 2000.
- [3] Raúl Joseph Schmuller, *Aprendiendo UML en 24 horas*, Prentice Hall.
- [4] Jerome Louvel; Thierry Boileau. *Restlet in Action*, Versión 5, 2011, MEAP Edition.
- [5] Restlet, *El principal Framework RESTful web para Java* , <http://www.restlet.org/>
- [6] JavaDoc Restlet, *public final class MediaType*, <http://www.restlet.org/documentation/2.0/jse/api/org/restlet/data/MediaType.html>
- [7] Agustín Froute Quintas. *JAVA 2 Manual de usuario y tutorial*, 5ª Edición, 2008, RA-MA Editorial.
- [8] Francisco Javier Ceballos. *JAVA 2 Curso de programación*, 4ª Edición, 2010, RA-MA Editorial.
- [9] Cadenhead, Rogers. *Java 6*, ANAYA Multimedia, 2008.
- [10] Eckel, Bruce. *Piensa en Java*, 4ª Edición, Pearson Educación, 2007.
- [11] ,Fernando Berzal Galiano, *Fundamentos de programación y principios de diseño*, <http://elvex.ugr.es/decsai/java/pdf/2B-Java.pdf> , <http://elvex.ugr.es/decsai/java/index.html>.
- [12] Larman, Craig. *UML y patrones : una introducción al análisis y diseño orientado a objetos y al proceso unificado*, Prentice Hall, 2003.
- [13] Mario G., Piattini; Jose A., Calvo Manzano; Joaquín, Cervera; Luis, Fernández. *Análisis y Diseño de Aplicaciones Informáticas de Gestión, Una perspectiva de Ingeniería del Software*, Ra-Ma, 2003.

- [14] Jennifer Niederst Robbins. *Creación y diseño web profesional*, 2008, ANAYA Multimedia.
- [15] Graig Cleaveland. *Program generators with XML and Java*, 2001, Prentice Hall PTR.
- [16] Fernanda V. Biegas, *MIT Media Lab*, Cambridge MA 02139 USA, fviegas@media.mit.edu.
- [17] Martín Wattenberg, *IBM Research*, Cambridge, MA 02142 USA, mwatten@us.ibm.com).
- [18] Kushal Dave, *IBM Research*, Cambridge, MA 02142 USA, kdave@us.ibm.com.
- [19] Licda Romelia Rodríguez, *Diagrama de Gantt*, <http://romelia.comli.com/files/TecnicasGANTT,PERTyCPM.pdf>
- [20] History Flow, *Herramienta para la visualización dinámica*, [http://www.research.ibm.com/visual/projects/history\\_flow/](http://www.research.ibm.com/visual/projects/history_flow/)
- [21] Prefuse, *Kit de herramientas de visualización Prefuse*, <http://prefuse.org/>
- [22] Applet, *Componente de una aplicación*, <http://es.wikipedia.org/wiki/Applet>
- [23] Gephi, *Visualización interactiva y plataforma de exploración de todo tipo de redes y sistemas complejos, dinámicos y gráficos jerárquicos*, <http://gephi.org/>
- [24] JVM, *Máquina Virtual de Java*, [http://es.wikipedia.org/wiki/M%C3%A1quina\\_virtual\\_Java](http://es.wikipedia.org/wiki/M%C3%A1quina_virtual_Java)
- [25] Richard Fearn, *Subversion plugin for History Flow*, <http://richardfearn.wordpress.com/2006/09/23/subversion-plugin-for-history-flow/>